drs. S.K.T. Boersma en ir. T. Hoenderkamp

SIMULATIE

een moderne methode van onderzoek



ACADEMIC
SERVICE

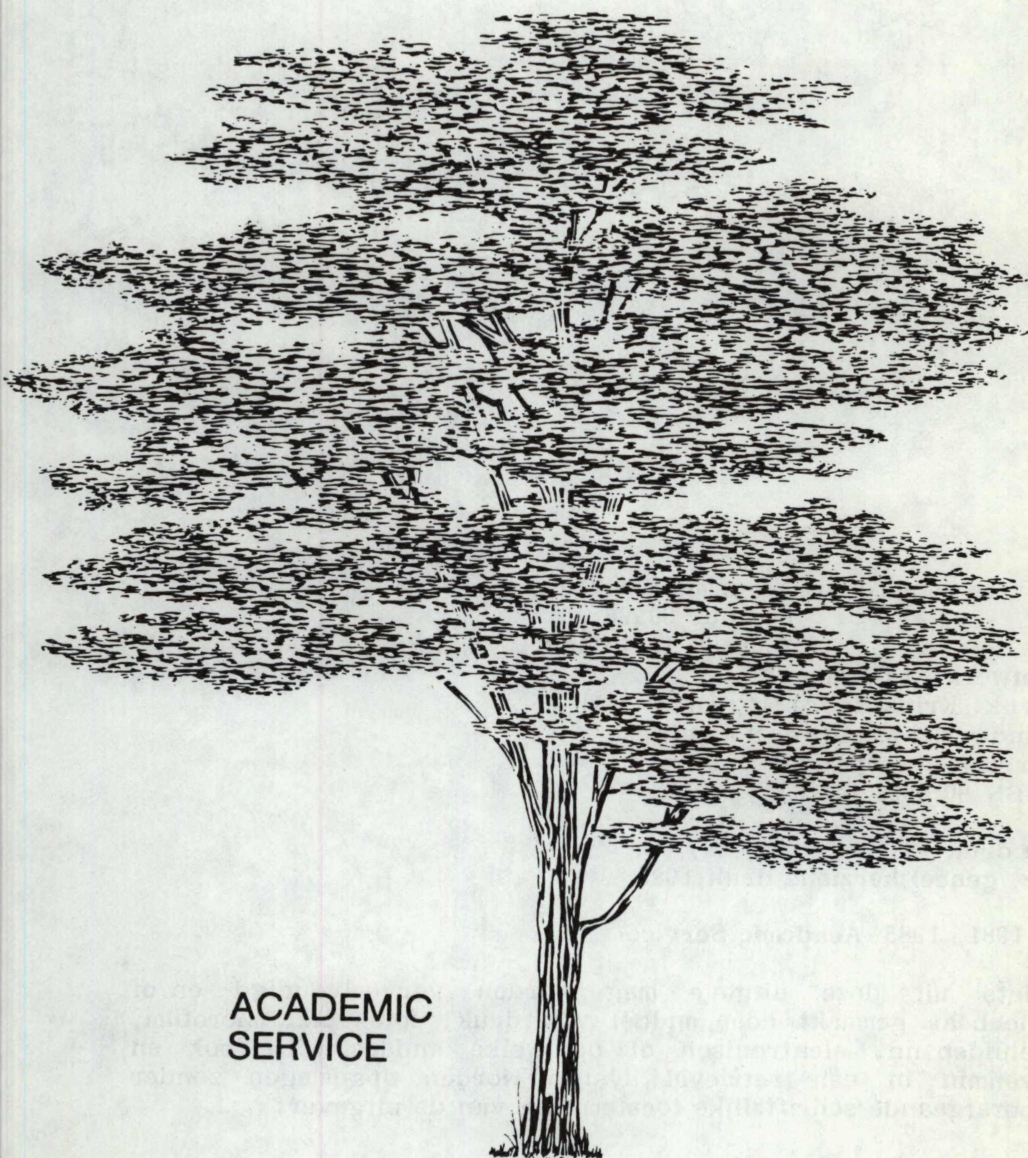
SIMULATIE

een moderne methode van onderzoek

drs. S.K.T. Boersma en ir. T. Hoenderkamp

SIMULATIE

een moderne methode van onderzoek



ACADEMIC
SERVICE

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Boersma, S.K.T.

Simulatie : een moderne methode van onderzoek / S.K.T. Boersma, T. Hoenderkamp. - Den Haag : Academic Service. - Ill.
ISBN 90-6233-140-8
SISO 365 SVS 8.12.3 UDC 681.3.068
Trefw.: informatica.

Uitgegeven door: Academic Service
Postbus 96996
2509 JJ Den Haag
Zetwerk: ATS, Lopik
Druk: Krips Repro Meppel
Bindwerk: Meeuwis, Amsterdam
Omslagontwerp: JAM Gauw
ISBN 90 6233 140 8

1e druk 1981
2e, geheel herziene druk 1985

Voorwoord bij de 2e Druk

In deze tijd van micro's, mini's en van supercomputers worden de toepassingsmogelijkheden van deze apparatuur steeds groter. Behalve voor snelle verwerking van massale hoeveelheden gegevens worden computers in steeds ruimere mate toegepast voor besluitvorming en beleidsvoorbereiding. De beslissingsproblemen waarvoor computers worden gebruikt zijn bijvoorbeeld personeelsplanning, ruimtelijke planning, kasvoorraad- en kredietbewakingsvraagstukken en produktieplanning. Deze problemen doen zich voor in allerlei soorten van organisaties: banken, ziekenhuizen, overheid en industriële organisaties. Het inzicht groeit, dat voor de oplossing van de genoemde beslissingsvraagstukken methoden nodig zijn die met inschakeling van computers ruime mogelijkheden bieden voor beslissingsondersteuning.

Simulatie is zo'n methode van onderzoek waarmee meer inzicht in complexe beslissingsproblemen kan worden verkregen. Simulatie geeft ons de mogelijkheid om onze kennis te vermeerderen, zowel ten aanzien van de structuur van grote en ingewikkelde systemen, als van oplossingen van de problemen. Ook kan simulatie ter verbetering van de communicatie een nuttig hulpmiddel zijn.

Op grond van de genoemde mogelijkheden is simulatie bij uitstek geschikt voor het maken van Decision Support Systems en van expertsystemen. Door deze systemen met gebruik van simulatie te bouwen, zal de effectiviteit van de besluitvorming kunnen toenemen. Met name in een vage of 'fuzzy' beslissingsomgeving kan zo het beslissingsproces met een geautomatiseerd informatiesysteem worden verbeterd.

VOOR WIE IS DIT BOEK BEDOELD?

Het boek is ontstaan uit collegedictaten voor studenten van de Interfaculteit Bedrijfskunde Groningen en voor HEAO-BI-studenten. Deze groepen worden in hun opleiding geconfronteerd met het vak Simulatie. Zij zijn ook in belangrijke mate de afnemers van de eerste druk van dit boek geweest. Daar-naast wordt het boek gebruikt bij HTS-en en economisch geöriënteerde opleidingen in het WO. Gezien de reacties heeft het boek ook ingang gevonden bij praktijkmensen. Hieruit blijkt, dat het boek zowel geschikt is

voor onderwijs aan studenten als voor zelfstudie. Wij zijn van mening, dat het boek behalve voor de genoemde studie-richtingen: Bedrijfskunde, Economie en Techniek ook geschikt is voor studenten uit de Informatica en de Sociale Wetenschappen. Door de verscheidenheid van oefeningen kan met het boek in uiteenlopende studierichtingen worden gewerkt.

DE OPBOUW VAN HET BOEK

Het boek is zodanig opgebouwd dat de lezer na bestudering van hoofdstuk I reeds aan de slag kan. In dit eerste hoofdstuk wordt namelijk een eenvoudig voorbeeld in extenso uitgewerkt. Na bestudering van dit hoofdstuk bevelen we de lezing van hoofdstuk IX aan waarin een tweetal cases (werkstukken van studenten) wordt behandeld.

De verschillende fasen van een simulatie-onderzoek worden in de hoofdstukken II-VI nader uitgewerkt. De hoofdstukken II en III gaan over modelbouw en hoofdstuk IV over het experimenteel ontwerp. Onder de titel 'Van model naar programma' worden in hoofdstuk V richtlijnen gegeven voor het maken van computerprogramma's. Ook worden in dit hoofdstuk voorbeelden van computerprogramma's in BASIC, Pascal en SIMULA gegeven. In hoofdstuk VI wordt uitgebreid ingegaan op het onderwerp Validatie.

Hoofdstuk VII is nieuw en behandelt het onderwerp Decision Support Systems. Behalve aan de vraag 'Wat zijn Decision Support Systems' wordt aandacht geschonken aan technische eisen, aan prototyping en aan een methode voor de bepaling van de waarde van dit soort systemen. Hoofdstuk VIII handelt over de bestudering van een simulatieproject.

Naast hoofdstuk IX met de twee uitgewerkte cases is een hoofdstuk met opgaven opgenomen. Met behulp hiervan kunnen simulatieprojecten in een onderwijsomgeving worden nagebootst (simulatie van de simulatie). Tenslotte zijn er twee appendices, één over statistische technieken en één over het programmeren van wachtrijen.

DANKWOORD

De auteurs zijn dank verschuldigd aan allen die op de één of andere wijze hebben bijgedragen aan de realisatie van dit boek. In het voorwoord bij de eerste druk noemden wij Prof.Dr. A. Bosman en Robert E. Shannon als de geestelijke vaders. Bij deze druk willen we hieraan met name toevoegen Drs. H. Alblas en P. Wagner. Beiden hebben in belangrijke mate bijgedragen tot deze hopelijk verbeterde versie. Naast deze met naam genoemden hebben anderen, collega's die ons hun kritieken hebben gegeven, en veel studenten in enigerlei vorm waarde toegevoegd. Opnieuw willen we Mevrouw Adele Hendriks

bedanken voor de vertaling van de geschreven tekst in dit typoscript. Bedankt daarvoor!

Ook nu houden we ons aanbevolen voor kritische kanttekeningen.

Glimmen januari 1985 S.K.Th. Boersma

Voorburg T. Hoenderkamp

Ten Geleide

Ter gelegenheid van de eerste druk heb ik opgemerkt dat het verschijnen van die publicatie over simulatie om verschillende redenen moest worden toegejuicht.

Twee van die redenen wil ik hier nog eens herhalen. Allereerst de kwaliteit van de publicatie. Ik was en ben van mening dat de auteurs er op voortreffelijke wijze in zijn geslaagd een goede combinatie te vinden tussen de complexiteit van het onderwerp en de voor het onderwijs noodzakelijke duidelijkheid van presentatie. Uit het feit dat een tweede druk noodzakelijk is drie jaren na het verschijnen van de eerste blijkt dat ook andere docenten deze mening zijn toegedaan.

Mijn tweede reden was dat de auteurs sterk het belang van simulatie als methode van onderzoek beklemtonen. De ontwikkeling in verschillende gebieden gedurende de laatste jaren toont aan dat deze stellingname correct is. Ik wil in dit verband wijzen op het gebruik van simulatie in decision support en expert systemen en het gebruik dat op grote schaal van zogenaamde decision support generatoren wordt gemaakt, die op microcomputers in de vorm van spread sheet programma's voorkomen. Voor het gebruik van deze nieuwe faciliteiten is het bijzonder gewenst dat men beschikt over enige kennis van simulatie. Simulatie kan worden beschouwd als een rode draad die kan worden gebruikt om relaties te leggen tussen de veelheid van toepassingen die de laatste jaren zijn ontwikkeld. Terecht, besteden de auteurs in deze nieuwe druk enige aandacht aan enkele van deze nieuwe ontwikkelingen.

De belangstelling voor simulatie en voor het onderhavige boek blijkt niet alleen te komen uit het hogere beroepsonderwijs, maar ook uit het wetenschappelijk onderwijs. Met dit feit en het verschijnen van deze tweede druk wil ik de auteurs en de uitgever gaarne complimenteren en de hoop uitspreken dat een derde druk 'binnenkort' weer noodzakelijk zal zijn.

Groningen, januari 1985 A. Bosman

Toen de Heer de aarde schiep en de mensen om erop te leven - een onderneming die, volgens de moderne wetenschap, nogal wat tijd in beslag nam - kan ik mij voorstellen dat hij bij zichzelf als volgt redeneerde: "Als ik alles voorspelbaar maak, zullen deze menselijke wezens, die ik heb toebedeeld met een vrij goed stel hersenen, ongetwijfeld leren alles te voorspellen en zij zullen dan geen motief hebben om ook maar iets uit te voeren, omdat zij zullen inzien dat de toekomst volkomen bepaald is en niet door enige menselijke handeling kan worden beïnvloed. Van de andere kant, als ik alles onvoorspelbaar maak, zullen zij gaandeweg ontdekken dat er geen rationele basis is voor welke beslissing dan ook, en zij zullen dan, net zoals in het eerste geval, geen motief hebben om ook maar iets uit te voeren. Geen van beide schema's zou zinvol zijn. Ik moet daarom een mengsel scheppen van beide. Zij zullen dan, bij veel andere, de belangrijke taak hebben te ontdekken hoe het zit.

E.F. Schumacher

citaat uit: 'Hou het klein:
een economische studie waarbij
de mens weer meetelt.'

Inhoudsopgave

HOOFDSTUK I

Beginselen van Simulatie

I.1 INLEIDING

In dit hoofdstuk willen we de lezer aan de hand van een algemeen inleidend verhaal en van een uitgewerkt voorbeeld een indruk geven van de facetten die bij een simulatie-onderzoek aan de orde komen. In paragraaf 2 wordt in grote lijnen aangegeven wat bedoeld wordt met simulatie als methode van onderzoek. De verschillende fasen van zo'n onderzoek worden besproken alsmede de samenhang tussen deze fasen. Hoewel een voorbeeld ter toelichting wordt gebruikt, zal het geheel nog als vrij theoretisch en complex worden ervaren. Daarom behandelen we in § 3 een heel eenvoudig en zeer praktisch probleem: de haringkar.

De eigenaar van een haringkar vermoedt, dat veel potentiële klanten weglopen omdat ze te lang moeten wachten. Om na te gaan of dit probleem door een extra bediende kan worden opgelost, wordt een eenvoudig simulatiemodel gebouwd. We behandelen bij de uitwerking van dit model alle fasen van een simulatie-onderzoek. Hierdoor hopen we te bereiken, dat de lezers al vrij snel zelf praktisch aan de slag kunnen. Simulatie kan volgens ons alleen goed worden geleerd als praktische voorbeelden zelfstandig op de computer worden uitgewerkt. Daarom ook zijn aan het slot van dit hoofdstuk in paragraaf 4 enkele eenvoudige opgaven opgenomen die zich goed lenen voor uitwerking met gebruik van de computer.

I.2 SIMULATIE IN VOGELVLUCHT

Bij simulatie moet aan nabootsing worden gedacht. Situaties uit de werkelijkheid worden op de één of andere manier nagebootst om zodoende kennis over of inzicht in die werkelijkheid te krijgen. De eenvoudigste manier om dit duidelijk te maken is een voorbeeld.

Veronderstel, dat we willen weten hoe groot de kans is om met twee dobbelstenen totaal twee ogen te gooien. De oplossing van dit eenvoudige probleem is 1/36; er bestaat namelijk slechts één combinatie van de 36 mogelijke combinaties die het gevraagde resultaat levert. De oplossing op grond van zo'n redenering wordt een analytische oplossing genoemd. Op grond van theoretische veronderstellingen wordt de uitkomst bepaald.

Een andere, meer praktische manier om de uitkomst vast te stellen is met de dobbelstenen te gooien en het resultaat van de worp te noteren. Als we dit een groot aantal keren doen (bijvoorbeeld 1000 keer) en we berekenen na afloop de relatieve frekwentie van het gevraagde resultaat dan zal deze in de buurt van de theoretische oplossing 1/36 moeten liggen. Het resultaat is dan bepaald door middel van een steekproefexperiment. Zo'n steekproefexperiment is een vorm van simulatie.

In dit boek zullen we ons overigens met deze vorm van simulatie, ook wel aangeduid als Monte Carlo-simulatie niet zo uitgebreid bezig houden. We komen er wel op terug, onder meer in hoofdstuk III, maar simulatie als experimenteertechniek is voor ons slechts een onderdeel van simulatie als onderzoekmethode. Dit hoofdonderwerp zullen we nu eerst wat uitgebreider inleiden.

In veel situaties ziet men zich geplaatst voor problemen waarvan niet duidelijk is hoe ze moeten worden opgelost. Soms is het zelfs niet duidelijk hoe men het probleem moet formuleren. In zulke gevallen zal men met vallen en opstaan structuur in het probleem proberen te vinden en tot oplossingen zien te komen. Hierbij zal vaak van ervaringsregels gebruik worden gemaakt. Doorgaans zijn die regels niet goed onderbouwd, ze zijn van een sterk kwalitatief en subjectief karakter.

Van oudsher heeft men in organisaties geprobeerd meer vat te krijgen op deze vorm van besluitvorming door van kwantitatieve technieken gebruik te maken. Na de 2e wereldoorlog is de operations research (OR) ontstaan. Dit vakgebied houdt zich bezig met de kwantitatieve ondersteuning van besluitvorming. De OR-technieken richten zich met name op toewijzings-, wachttijd-, voorraad- en volgorde vraagstukken. Met behulp van de techniek van lineaire programmering kan bijvoorbeeld een bepaalde groep toewijzingsproblemen worden opgelost. Het vele rekenwerk wordt door de computer snel en betrouwbaar verzorgd.

Toch hebben de meeste van deze technieken uit de operations research hun beperkingen. Zo is het altijd noodzakelijk het op te

lossen probleem om te zetten in wiskundige vergelijkingen. Er moet een mathematisch model worden gemaakt dat met behulp van een voor dit doel geschikte analytische techniek opgelost dient te worden. Veel problemen lenen zich hiervoor evenwel niet, omdat ze te ingewikkeld of slecht gestructureerd zijn. Voorbeelden zijn voorraadbeheersing, wachttijdproblemen met meerdere loketten, produktieplanning en investeringsselectie. Tot voor enkele jaren was het dan alleen in heel beperkte zin mogelijk ondersteuning te bieden. Door de opkomst van simulatie zijn de mogelijkheden tot het oplossen van de geschetste problemen aanzienlijk uitgebreid.

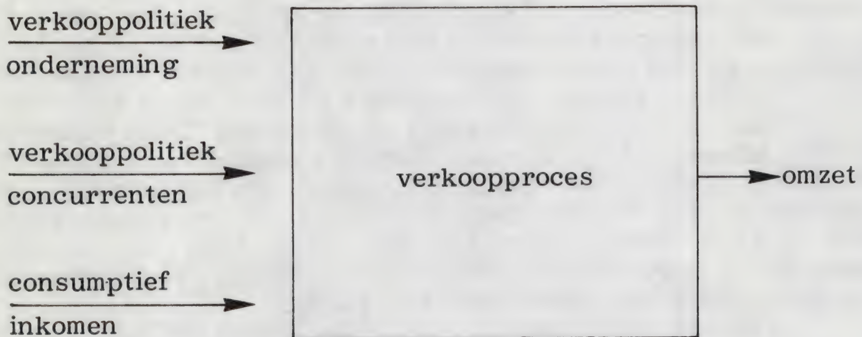
Een opvallend verschil tussen simulatie en de hierboven genoemde operations research-technieken is, dat men bij simulatie het model van de te onderzoeken werkelijkheid niet in wiskundige relaties behoeft te formuleren. Men probeert de werkelijkheid (het reële systeem) *na te bootsen* in de vorm van een proces of van een reeks gebeurtenissen. Zo kunnen bijvoorbeeld de groei van een gewas, het bedieningsproces in een winkel of het op de markt brengen van een nieuw produkt worden nagebootst. Hiertoe wordt de werkelijkheid eerst zo goed mogelijk beschreven in een model om daarna met dit model experimenten uit te voeren. Deze experimenten hebben allereerst tot doel werkelijkheid en model zo veel mogelijk met elkaar in overeenstemming te brengen. Als dit eenmaal is gelukt kan aan de hand van het model meer inzicht in het gedrag van het reële systeem worden verkregen. Daartoe moet het model dan wel eerst in een computerprogramma worden omgezet, zodat gemakkelijker met het model geëxperimenteerd kan worden.

Om één en ander beter toe te lichten maken we gebruik van een voorbeeld. In dit voorbeeld gaat het om een onderneming die verschillende produkten op de markt brengt. Met deze produkten dient een zo groot mogelijke omzet te worden gerealiseerd. Daartoe kan van verschillende verkoopmethoden gebruik worden gemaakt. Om het eenvoudig te houden beperken we ons tot twee soorten verkooppolitiek: verkooppolitiek I en verkooppolitiek II. De leiding van de onderneming is geïnteresseerd in de vraag: Welke van de twee soorten verkooppolitiek zal de omzet het meest stimuleren?

Een antwoord op deze vraag is door praktische uitvoering misschien wel te geven, maar deze methode is kostbaar. Het liefst zal men tevoren weten welke verkooppolitiek het meeste succes zal opleveren. Simulatie is één van de mogelijkheden waarmee relevante informatie vóór de definitieve beslissing kan worden verkregen. Het verkoopproces wordt gedurende bepaalde tijd nagebootst. Men doet dit éénmaal bij verkooppolitiek I en nogmaals bij verkooppolitiek II onder verder gelijkblijvende omstandigheden. Uit dit 'doorspelen' van het verkoopproces verkrijgt men twee reeksen omzetcijfers, die behoren bij de verkooppolitiek I respectievelijk politiek II. De reeksen cijfers worden statistisch vergeleken, dat wil zeggen dat men de cijfers aan een aantal statistische toetsen onderwerpt om te kijken of er een significant verschil is tussen de invloed op de omzet van de beide verkooppolitieken. Dan kan men uit deze waarnemingen conclusies trekken omtrent de 'beste' verkooppolitiek.

Zowel het experimenteren alsook de statistische analyse gebeurt met behulp van de computer. Hiervoor is het in de eerste plaats nodig dat we beschikken over een *model* van de onderneming. Het bouwen van zo'n model is de belangrijkste en wellicht ook de moeilijkste activiteit binnen een simulatieproject. Welke variabelen nemen we op en welke laten we weg? Welke relaties moeten worden opgenomen? Lastige vragen waarop de antwoorden niet altijd voor de hand liggen.

We zullen in dit voorbeeld verder veronderstellen, dat de omzet beïnvloed wordt door de verkooppolitiek van de onderneming zelf, de verkooppolitiek van de concurrentie en door het consumptieve inkomen. In een eenvoudig schema kunnen deze keuzes als volgt worden weergegeven:



Figuur I.2a

Een zo mogelijk nog belangrijker vraag en minstens even moeilijk te beantwoorden, is de vraag *hoe* de omzet afhangt van de beïnvloedingsfactoren. Deze afhankelijkheid is de verzameling relaties in ons ondernemingsmodel. Hoewel - en dit is het verschil met mathematische analysemethoden - deze relaties *niet* in de vorm van wiskundige vergelijkingen behoeven te worden gedefinieerd, dient men wel uitspraken te doen over de ordening van de relaties. Dit wil zeggen, dat wel de volgorde in de zin van: 'gaat vooraf aan'. of 'volgt op' moet worden aangegeven, of dat procesrelaties als 'X moet leveren aan Y' omschreven moeten worden.

Als eenmaal een model is geformuleerd en geaccepteerd, dan is de eerste fase van het simulatie-onderzoek achter de rug. In deze fase, ook wel de inductiefase genoemd, is een aantal stappen te onderscheiden:

1.	de probleemspecificatie en de specificatie van de doelstellingen van het model
2.	het formuleren van een kwalitatief model
3.	het specificeren van een voorlopige verzameling variabelen en parameters
4.	het specificeren van de relaties tussen de variabelen in de vorm van hypothesen
5.	met behulp van de waarnemingsgegevens formuleren van een kwantitatief model, waarbij de relaties van een inhoud worden voorzien.

Wanneer we - met vallen en opstaan - deze stappen tot een bevredigend einde hebben gebracht, betekent dat geenszins het einde van het werk. Als namelijk de relaties in een kwantitatieve vorm zijn beschreven (bijvoorbeeld in de vorm van een computerprogramma), betekent dat niet dat men nu over een verklaring beschikt. Dat is een fout die veel gemaakt wordt. Het bouwen van een model is geen doel op zich, maar een middel. Het model moet nog opgelost worden. Simulatie is een heuristische methode. Dit houdt in, dat met behulp van een model gezocht wordt naar een aanvaardbare oplossing van het probleem waarvoor een model is geformuleerd. Dit zoekproces gebeurt in de volgende fase van het onderzoek: het experimenteren met het model.

In deze tweede fase staat dus het experiment centraal. Experimenteren doen we evenwel niet in het wilde weg: we bereiden de experimenten zorgvuldig voor.

In de eerste plaats zullen we experimenteren met het doel informatie te krijgen omtrent de juistheid van het model. De analyse van de verkregen uitkomsten behoort bij de validatie, één van de fasen uit de modelcyclus. Daar de keuze van de validatiemethoden wel de manier bepaalt waarop de experimenten worden ingericht, dient men deze reeds tevoren te hebben overdacht.

Wanneer het vertrouwen in de juistheid van het model op een acceptabel niveau is gekomen, zullen we *in de tweede plaats* experimenten uitvoeren om uitspraken te kunnen doen over het gedrag van het reële systeem. Hierdoor proberen we inzicht te krijgen in de probleemsituatie. De informatie die wij nodig hebben om gefundeerde uitspraken te kunnen doen, wordt niet zonder meer door willekeurige experimenten voortgebracht.

Nauw hierbij aansluitend is *tenslotte* de vraag naar het toekomstig gedrag van het systeem interessant. Wij beoordelen modellen vaak naar de voorspellende capaciteiten ervan. Immers de problemen liggen dikwijls op het terrein van beslissingen die betrekking hebben op de toekomst.

Hoe de voorbereiding van de experimenten, ook wel aangeduid als het experimenteel ontwerp, gebeurt behandelen we op deze plaats niet. In hoofdstuk IV gaan we op dit onderwerp nader in. Wel willen

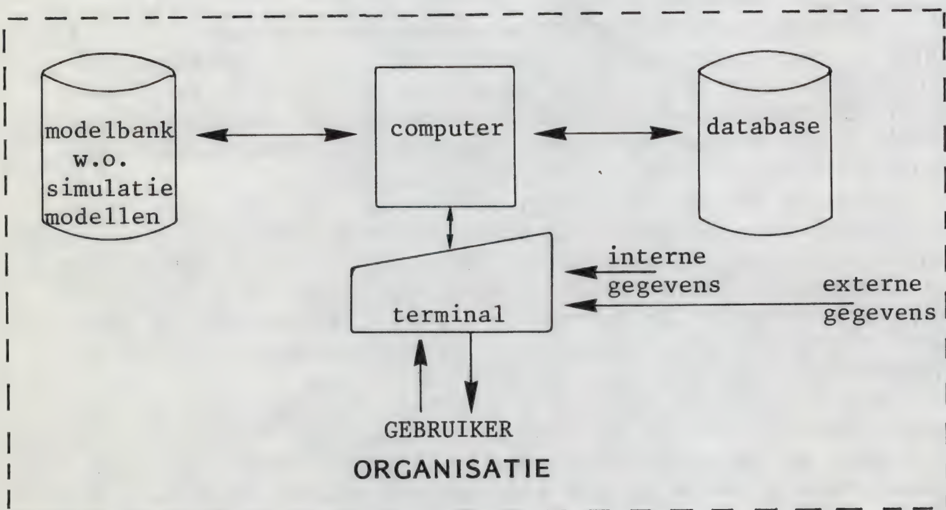
we er nu reeds op wijzen dat de vorm van het experimenteren sterk afhangt van de doelstellingen van de simulatie. Veel voorkomende mogelijkheden zijn:

- het vergelijken van alternatieven (verkooppolitiek I of verkooppolitiek II);
- het bepalen van de invloed van verschillende variabelen;
- het bepalen van de onderlinge afhankelijkheid van de variabelen (zal onze verkooppolitiek de verkooppolitiek van de concurrentie beïnvloeden?);
- de invloed op de responsevariabelen van veranderingen in de waarde van de exogene variabelen (gevoeligheidsanalyse).

Nadat met behulp van het experimenteren het model gevalideerd is en de opdrachtgevers en gebruikers voldoende vertrouwen hebben gekregen in het model, zal vaak de derde fase volgen, namelijk: de invoering van het model om het in de praktijk te gaan gebruiken.

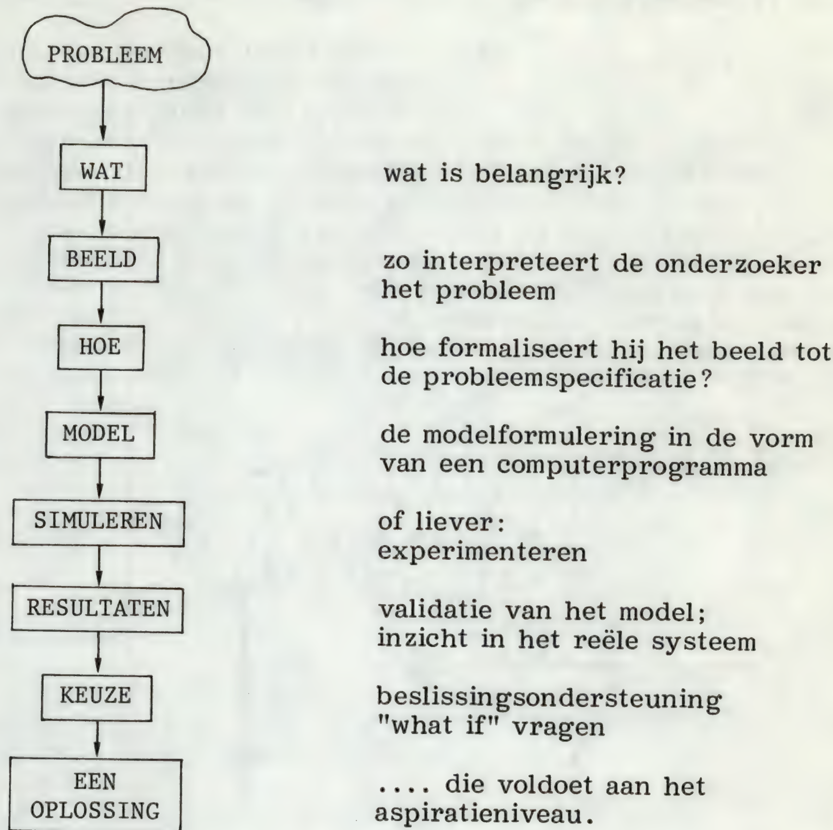
Wanneer ertoe wordt overgegaan om het model te gebruiken als een gereedschap bij het voorbereiden en nemen van beslissingen ("management-tool"), dan wordt het simulatiemodel opgenomen in het concrete informatiesysteem.

Het model kan deel uitmaken van een modelbank. Gebruikers kunnen het model aan de terminal 'oproepen' en ermee experimenteren, waarbij gebruik gemaakt kan worden van de bedrijfsgegevens uit de data-base, eventueel nadat deze bewerkingen hebben ondergaan (statistische bewerkingen, aggregatie, e.d.). Dit type modellen staat bekend onder de naam decision support systems (DSS). In een schema kan deze situatie als volgt worden weergegeven:



Figuur I.2b

In deze paragraaf is ingegaan op de vraag: wat is simulatie? Gedeeltelijk aan de hand van een voorbeeld zijn de fasen van een simulatieproject in grote lijnen beschreven. Voor de duidelijkheid presenteren we de hoofdlijnen van het betoog in figuur I.2c.



Figuur I.2c

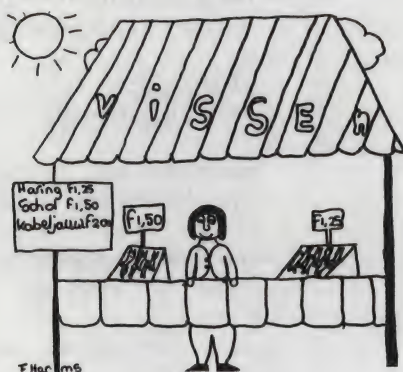
In de volgende paragraaf werken we een eenvoudig probleem volgens dezelfde opbouw als in figuur I.2c geheel uit.

I.3 HET SIMULATIE-ONDERZOEK IN FASEN; EEN VOORBEELD

I.3.1 Probleembeschrijving

Op de hoek van een straat in mijn buurt staat iedere avond van ca. vijf tot ongeveer half zeven een haringkar. Als je op de eerste dagen van de week een boodschap doet bij de kar is het in het algemeen rustig en hoef je nauwelijks te wachten. Naarmate de week voortschrijdt, wordt het drukker bij de kar en op vrijdag en zaterdag moet je meestal nogal lang wachten. Deze wachttijden wekken enige ergernis bij de klanten op. Ik heb daarom aan de haringman gevraagd, of hij op deze drukke dagen niet een hulp zou kunnen nemen. "Jawel," was zijn antwoord, "als ik maar wist dat het zou helpen. Om nu eerst iemand in dienst te nemen en hem dan later weer te moeten ontslaan als blijkt dat het toch niet helpt, vind ik onverantwoord."

*)



Een vrij eenvoudig probleem, dat met eenvoudige hulpmiddelen kan worden opgelost. Hoewel computersimulatie in dit geval als 'het loslaten van een olifant op een muis' kan worden beschouwd, is het probleem wel geschikt om een aantal basisprincipes van simulatie toe te lichten.

I.3.2 Probleemformulering en doelstelling

Hierboven hebben we een korte beschrijving gegeven van het probleem van de haringverkoper. Als onderdeel van de probleem-

---

*) De tekeningen in dit boek zijn gemaakt door de 5-6 groep van de Montessorischool 'Nieuw Vreugd en Rust' te Voorburg.

formulering moet deze beschrijving enerzijds iets zeggen over de bestaande situatie en anderzijds aangeven wat de gewenste toestand dient te zijn.

Ten behoeve van een beschrijving van de bestaande toestand kan gebruik worden gemaakt van technieken als directe waarneming, enquête en interview. Het is voor dit probleem bijvoorbeeld denkbaar, dat zowel met de haringman zelf als met een aantal klanten gesprekken worden gevoerd. Hierbij moet eveneens aandacht worden geschonken aan de gewenste toestand.

Dit vooronderzoek dient in ieder geval ook te leiden tot een omschrijving van de doelstelling van het eigenlijke onderzoek. De doelstelling van het project 'haringkar' zou omschreven kunnen worden als:

Onderzoek de mogelijkheden om de wachttijden van de klanten op drukke dagen terug te brengen tot een aanvaardbaar gemiddelde.

Hoewel deze doelstelling op het eerste gezicht acceptabel lijkt, is hij op een aantal punten toch wel te vaag. We zullen dit toelichten aan de hand van een aantal punten.

In de eerste plaats krijgen we te maken met de vraag: wat is een aanvaardbaar gemiddelde?

Deze vraag zou beantwoord kunnen worden door middel van een kleine enquête onder de vaste klantenkring. Uit deze enquête zou kunnen blijken, dat de meeste klanten een gemiddelde wachttijd van maximaal 5 minuten aanvaardbaar achten. Op zichzelf kan dit gegeven voldoende zijn voor de gestelde vraag, maar ... pas op! Het zou kunnen zijn, dat het hier gaat om 80% van de klanten die slechts 20% van de omzet voor hun rekening nemen. De andere 20% van de klanten, die dus 80% van de omzet van de haringverkoper verzorgen zouden misschien al ontevreden worden bij een wachttijd van één minuut.

Ook is het mogelijk dat de enquête als resultaat geeft dat de klanten het helemaal niet erg vinden om te wachten. In zo'n geval kan een verder onderzoek dus achterwege blijven.

Een tweede opmerking heeft te maken met de zinsnede: Onderzoek voor de haringverkoper de mogelijkheden ..... Dit deel van de doelstelling is ruim, zowel ten aanzien van de methoden van onderzoek als van de mogelijke oplossingen.

Gegeven de context van het probleem zullen wij aannemen dat als methode van onderzoek simulatie wordt gebruikt en als criteria voor de oplossing van het probleem: de gemiddelde wachttijd en de gemiddelde lengte van de wachtrij. Bovendien moeten uitspraken worden gedaan omtrent de spreiding rond het gemiddelde, omdat het gemiddelde alleen een te beperkte maatstaf is.

Een derde opmerking heeft betrekking op de kosten van een onderzoek. Wat willen we bereiken en wat hebben we voor dit doel

over? Er moet een kosten-baten-analyse voor het onderzoek worden gemaakt en deze moet aan de opdrachtgever(s) worden voorgelegd.

Ter ondersteuning van het opstellen van een kostenoverzicht kan gebruik worden gemaakt van de diverse fasen, die in een onderzoek zijn te onderscheiden en van diverse kostencomponenten, zoals personeelskosten en bij simulatie vooral programmerings- en computerkosten.

In veel gevallen zullen de uit een onderzoek voortvloeiende baten erg moeilijk zijn aan te geven. Dit geldt ook voor het probleem van de haringverkoper voor wie de baten vooral moeten voortkomen uit een daling van het aantal mensen dat vertrekt of niet komt wegens te lange wachttijden. Schattingen hierover zijn moeilijk te geven, maar op grond van bijvoorbeeld enquêtes en/of directe waarneming in combinatie met statistische technieken zeker niet geheel onmogelijk.

I.3.3 Het bouwen van het model

Het probleem laat zich als volgt in een eenvoudig schema weergeven:



Figuur I.3a

Door dit schema willen we aangeven, dat er klanten bij de haringkar aankomen, dat ze al of niet wachten op hun beurt, dat ze worden geholpen als ze aan de beurt zijn en dat ze na hun beurt weer vertrekken. Voor de constructie van het model is het van belang te weten wanneer de klanten aankomen, de concentratie van de aankomsten en de bestellingen van de klanten met de daarmee gepaard gaande bedieningstijden.

Om wat meer inzicht in het probleem te krijgen, kunnen we ook een beknopte beschrijving geven van de gebeurtenissen die zich rond de haringkar voordoen:

1. op een bepaald tijdstip komt er een klant bij de kar om vis te kopen;
2. de klant sluit zich aan bij de eventueel aanwezige wachtrij;
3. als de klant aan de beurt is, wordt hij geholpen, waarbij de bedieningstijd een belangrijk gegeven is;
4. nadat een klant is geholpen, verlaat hij de kar en de wachtrij schuift één op.

Ten behoeve van de doelstelling van de simulatie zijn o.a. gegevens over de aankomsttijd noodzakelijk. Deze gegevens kunnen worden verkregen door metingen ter plaatse uit te voeren. Deze metingen geven ons dan een indruk van de bestaande situatie.

Hoewel we deze metingen, zoals we nog zullen zien, heel goed kunnen gebruiken, zijn ze voor het onderzoek niet voldoende. Om het onderzoek te kunnen uitvoeren zullen we ook een indruk moeten krijgen van het bedieningsproces. We zullen als het ware de 'black box' moeten openen om zodoende van de inhoud een indruk te krijgen en de relevante aspecten ervan te beschrijven. Juist door dit bedieningsproces te analyseren zullen we achter de zogenaamde *stuurvariabelen* (of: instrumentele variabelen) kunnen komen. De stuurvariabelen zijn die variabelen welke gebruikt kunnen worden ter beïnvloeding van de uitvoervariabelen.

In dit voorbeeld veronderstellen we één stuurvariabele: het aantal bedienden. Zoals bekend is dit aantal nu één en de vraag is of het zinvol is dit aantal tot twee uit te breiden.

Terzijde merken we op dat het bedieningsproces natuurlijk meer aspecten kent die van invloed zijn op de wachttijd. Zo zijn bijvoorbeeld ook het aantal artikelen, de voorbereidingstijd alvorens de artikelen ingepakt aan de klant aangeboden kunnen worden van invloed. Om het model niet al te ingewikkeld te maken laten we deze aspecten hier verder buiten beschouwing.

Een voor ons belangrijk kenmerk van een bediende is de bedieningstijd. Evenals voor de aankomsttijd zullen we om een inzicht te krijgen in de bedieningstijd metingen moeten doen.

Zowel de aankomsttijd als de bedieningstijd zijn kansvariabelen, ook wel aangeduid als stochastische variabelen. Om te kunnen simuleren zullen we voor deze variabelen over een frequentieverdeling met de bijbehorende parameters moeten beschikken.

De metingen die we uitvoeren aan de genoemde variabelen hebben tot doel een frequentieverdeling te kunnen opstellen en kenmerkende grootheden ($\equiv$ parameters) als rekenkundig gemiddelde *) en variantie te kunnen berekenen.

We zullen aannemen dat de gevonden empirische frequentieverdelingen vervangen kunnen worden door uit de statistiek bekende theoretische verdelingen. Op de achtergrond en de toetsing van deze veronderstelling zullen we in hoofdstuk III nader ingaan. We bespreken achtereenvolgens een drietal veronderstellingen.

---

*) Waar in dit boek verder gesproken wordt over het gemiddelde zonder nadere aanduiding wordt het rekenkundig gemiddelde bedoeld.

Veronderstelling 1

De aankomsttijd van de klanten kan beschreven worden door de kansdichtheid

$$f(x) = \lambda e^{-\lambda x} \qquad x > 0;\ \lambda > 0$$

waarin x een kansvariabele is die het tijdsinterval tussen twee aankomsten van een klant voorstelt. λ is de parameter van deze verdeling, die staat voor het gemiddeld aantal klanten dat per tijdseenheid aankomt. Als λ bekend is (bijvoorbeeld door metingen), kan voor elke x een kans worden berekend.

Deze kansdichtheid staat bekend als de negatief exponentiële verdeling die in nauwe relatie staat met de Poisson-verdeling. Als namelijk de aankomsten in een systeem volgens Poisson verdeeld zijn, dan is de tussenaankomsttijd (x) negatief exponentieel verdeeld.

De toepassing van dit kansmechanisme is alleen geoorloofd als aan de volgende punten is voldaan:

a. er komen eindig veel klanten binnen in een eindig tijdsinterval;
b. elke klant komt afzonderlijk binnen;
c. de klanten komen onafhankelijk van elkaar binnen;
d. de kans op het binnenkomen van een klant in een bepaald tijdsinterval is voor alle tijdsintervallen dezelfde.

De Poisson-verdeling kan, zoals bekend mag worden verondersteld, beschreven worden door de volgende formule:

$$f(k) = e^{-\lambda} \frac{\lambda^k}{k!} \qquad (k = 0,1,2,.....)$$

Het gemiddelde (de enige parameter van de Poisson-verdeling) is λ of beter λ eenheden per tijdseenheid. De gemiddelde tussenaankomsttijd is dus $1/\lambda$. Dit is dan tevens het gemiddelde van de negatief exponentiële verdeling.

Veronderstelling 2

Behalve het aankomstproces is voor de vaststelling van de gemiddelde wachttijd en de gemiddelde lengte van de wachtrij ook het proces van bediening van betekenis. Immers als de gemiddelde bedieningstijd groter is dan de gemiddelde tussenaankomsttijd, dan zal de wachtrij langer en langer worden.

Voor de bedieningstijd kan in veel gevallen ook een negatief exponentiële verdeling worden aangenomen. Ter afwisseling met veronderstelling 1 nemen we aan dat de bedieningstijd uniform verdeeld is.

De uniforme verdeling (ook wel continue, homogene of rechthoeksverdeling genoemd) is van toepassing op een continue variabele t (bijvoorbeeld de tijd) die binnen een bepaald interval (A, B)

$$f(t) = \frac{1}{B-A} \quad \text{voor} \quad A < t < B,$$

$$f(t) = 0 \quad \text{elders.}$$

De parameters van deze verdeling zijn:

$$\mu_{\underline{t}} = \tfrac{1}{2}(A + B)$$

$$\sigma^2_{\underline{t}} = \tfrac{1}{12}(B - A)^2 \ .$$

Veronderstelling 3

We nemen tenslotte aan dat de invloed van het aantal verkopers tot uitdrukking komt in de waarden van het interval (A, B) van de uniforme verdeling.

Zo kan voor één verkoper bijvoorbeeld gelden, dat

$$1 < t < 3$$

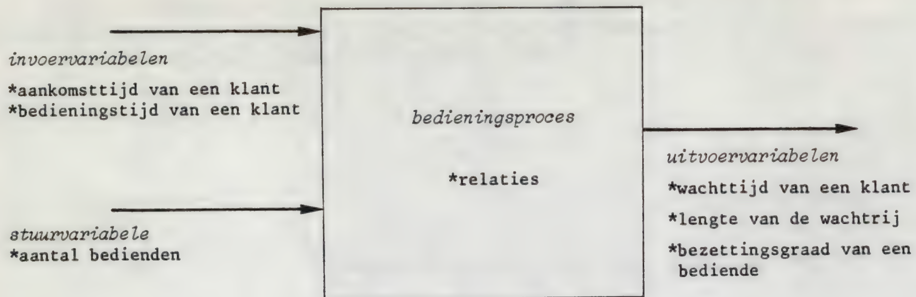
en voor twee verkopers

$$\tfrac{1}{2} < t < 2 \ .$$

We hebben nu voldoende ingrediënten om een beschrijving te kunnen geven van het model voor simulatie van de haringkar.
We doen dit met behulp van de volgende begrippen:
- invoervariabelen
- stuurvariabelen
- uitvoervariabelen
- relaties.

Hoewel deze begrippen nog niet behandeld zijn, nemen we aan dat de bedoeling ervan door de voorbeelden duidelijk zal worden. Een algemene behandeling van deze termen volgt in hoofdstuk II.

Voor de duidelijkheid geven we eerst een schets van het model.



Figuur I.3b

Ter aanvulling op deze schets is voor een volledige beschrijving de volgende informatie noodzakelijk.

Verdeling van de invoervariabelen

Zie hiervoor de veronderstellingen 1, 2 en 3.

Verlangde uitvoer (informatie) van het model

Ten aanzien van de *wachtrij*:
- maximale rijlengte;
- gemiddelde rijlengte.

Ten aanzien van de *klanten*:
- het gemiddeld aantal klanten per tijdseenheid in het systeem.

Ten aanzien van de *wachttijd*:
- de gemiddelde wachttijd en de standaardafwijking van de wachttijd (naast het rekenkundig gemiddelde kunnen ook de modus en de mediaan van belang zijn);
- een frequentieverdeling van de wachttijd.

Ten aanzien van de *bezetting*:
- de bezettingsgraad van het systeem, d.w.z. informatie over de tijd dat er geen klanten zijn ten opzichte van de totale tijd dat de kar bezet is.

Relaties (in de vorm van *aannamen* of *hypothesen*)

- Het aankomsttijdstip van een klant is een trekking uit een negatief exponentiële verdeling.
- De bedieningstijd van een klant is een trekking uit een uniforme verdeling.

- De rijdiscipline is FIFO (first in, first out).
- Na aankomst wordt de klant achterin de wachtrij geplaatst.
- Als de rij leeg is, roept de klant de haringman tot aktie.
- De wachttijd van een klant is het tijdsverschil tussen het moment van start van de bediening en het tijdstip van aankomst.
- Als de rij niet leeg is, haalt de haringman, nadat de vorige klant geholpen is, de eerste klant uit de rij, om deze te bedienen.
- Een aantal niet verder te specificeren relaties om de statistiek bij te houden (bijvoorbeeld de cumulatieve wachttijd wordt verhoogd, telkens als een klant wordt bediend).

Het is een goede gewoonte om, als dit gezien de complexiteit mogelijk is, het model 'met de hand' door te rekenen ('desk check'). Dit is vooral van belang om eventuele logische fouten reeds in een vroeg stadium op te sporen. Bij dit doorrekenen kan het beste van eenvoudige getallen gebruik worden gemaakt. Als hulpmiddel kan van een tabel als tabel I.1 gebruik worden gemaakt.

I.3.4 Een verdere uitwerking

Om het model door de computer te kunnen laten doorrekenen is het nodig, dat in het programma aankomsttijden en bedieningstijden worden gegenereerd. Naast de verdelingsfuncties zijn daarom parameters nodig. Deze parameters kunnen aan de hand van metingen bij de haringkar worden geschat. In het voorbeeld nemen we voor de gemiddelde tussenaankomsttijd twee minuten aan. De parameters van de uniforme verdeling voor de bedieningstijd zijn één minuut en drie minuten.

Het belangrijkste hulpmiddel voor het genereren van aankomst- en bedieningstijden zijn zogenaamde toevalsgetallen (Engels: random numbers). Dat zijn getallen die beschouwd worden als aselecte trekkingen uit een homogene verdeling op het interval (0,1). In sommige statistische handboeken zijn tabellen van deze toevalscijfers opgenomen. De methoden om m.b.v. de computer deze getallen te genereren worden in hoofdstuk III besproken.

Omdat elke verdelingsfunctie F(x) van een kansdichtheid gedefinieerd is op het interval (0,1) kunnen aselecte trekkingen uit zo'n kansdichtheid worden verkregen door de betrokken toevalscijfers gelijk te stellen aan F(x) en deze vergelijking op te lossen. We zullen dit illustreren aan de hand van de beide verdelingsfuncties in het voorbeeld van de haringkar.

Voor de exponentiële verdeling kan de verdelingsfunctie worden berekend uit:

$$F(x) = \int_0^x \lambda e^{-\lambda x} dx = -e^{-\lambda x}\Big]_0^x = 1 - e^{-\lambda x} \qquad (x > 0)$$

Als we een aselect getal r trekken dan kan een uitkomst van de kansvariabele x worden verkregen door op te lossen:

$$r = 1 - e^{-\lambda x} \rightarrow e^{-\lambda x} = 1 - r \rightarrow$$

$$x = -\frac{1}{\lambda} \ln (1-r) .$$

Voor $1/\lambda = \frac{1}{2}$ en voor de 5 aselecte getallen:

0,185; 0,234; 0,032; 0,785 en 0,973

krijgen we de volgende tijdsintervallen:

0,843; 0,726; 1,721; 0,121 en 0,013.

Voor de bedieningstijd hebben we een uniforme verdeling verondersteld. De verdelingsfunctie volgt uit:

$$F(t) = \int_A^t \frac{1}{B-A} dt = \frac{t-A}{B-A} \qquad (A<t<B) .$$

Op analoge wijze als hierboven kan worden afgeleid:

$t = (B-A)r + A$.

Als we nu bijvoorbeeld $A = 1$ en $B = 3$ nemen en als we de volgende 5 aselecte getallen genereren:

0,217; 0,981; 0,473; 0,021; 0,845

dan zijn de volgende bedieningstijden te berekenen:

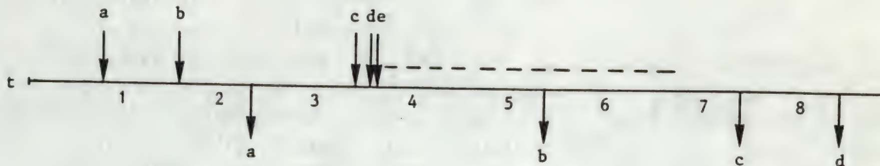
1,434; 2,962; 1,946; 1,042; 2,690.

Door deze tijden met elkaar in verband te brengen, krijgen we een indruk van de werking van ons systeem. In de volgende tabel is deze relatie gelegd.

klant	tussen-aankomst-tijd	tijdstip van aan-komst	bedienings-tijd	tijdstip van vertrek	wachttijd
a	0,843	0,843	1,434	2,277	1,434
b	0,726	1,569	2,962	5,239	3,670
c	1,721	3,290	1,946	7,185	3,895
d	0,121	3,411	1,042	8,227	4,816
e	0,013	3,454	2,690	10,917	7,463
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-

Tabel I.1

Ook kunnen we de gebeurtenissen op een tijdas uitzetten om zodoende snel een indruk te krijgen van de aankomsttijdstippen en de vertrektijdstippen van de klanten en daardoor van de wachttijden en de rijlengte.



Figuur I.3c

Als deze gegevens representatief zijn voor de werkelijke gebeurtenissen tijdens de 'spits' bij de haringkar is het duidelijk, dat er sprake is van een onaanvaardbare situatie voor de klanten. Teneinde voorstellen te kunnen doen om in deze situatie verbetering te brengen kan de in § I.3.7 voorgestelde werkwijze worden gevolgd. Op grond van de gemaakte berekeningen kan dan vervolgens aan de verkoper een advies worden uitgebracht.

I.3.5 Het schrijven van een computerprogramma

Alvorens het geformuleerde model te kunnen testen en met het model te kunnen experimenteren zullen we een computerprogramma moeten schrijven.

De omzetting van het model tot een goed computerprogramma kent een drietal fasen, namelijk:

a. een gestructureerde formele beschrijving van het model
b. de codering van de gestructureerde beschrijving in een computerprogrammeertaal
c. het testen van het programma.

<u>Ad a.</u> Als we spreken van een gestructureerde beschrijving van het model, dan bedoelen we daarmee, dat in het programma-ontwerp op een overzichtelijke wijze de logische opbouw van het probleem is weergegeven. Een aantal, elkaar aanvullende mogelijkheden hiervoor zijn:

1. een zogenaamde 'top-down' aanpak
2. gebruik maken van modules (modulair programmeren)
3. gebruik maken van de drie fundamentele besturingsstructuren.

In hoofdstuk V komen we op deze punten uitvoerig terug.

<u>Ad b.</u> Voor de codering kan de onderzoeker zich bedienen van een hogere programmeertaal. Meerdere programmeertalen zijn beschikbaar, zoals Pascal, FORTRAN, ADA of M-BASIC. Deze talen worden wel aangeduid als *proceduregerichte talen*. Zij beschikken over

standaardprocedures en de mogelijkheid tot het zelf maken van procedures waardoor een veelheid van problemen kan worden beschreven. Deze talen lenen zich dus voor een grote veelsoortigheid van vraagstukken. Voor simulatietoepassingen is het belangrijkste nadeel van deze talen dat de voor deze toepassingen noodzakelijke procedures door de programmeur zelf geschreven moeten worden. Dit nadeel hebben de speciaal voor simulatietoepassingen ontworpen talen als SIMULA, PROSIM en SIMSCRIPT niet. Zij beschikken over mogelijkheden om een simulatiemodel effectief en efficiënt om te zetten in een computermodel. Deze talen worden, omdat ze voor een bepaald soort van problemen zijn ontwikkeld, gerekend tot de *probleemgerichte talen*. Ook voor andere soorten van problemen zijn dergelijke talen gemaakt.

In hoofdstuk V zullen we enige aandacht besteden aan de volgens ons voor simulatietoepassingen meest geschikte taal: SIMULA.

Ondanks de voordelen van deze simulatietalen zijn ze (nog?) slechts beperkt beschikbaar op computers van organisaties. Hierdoor is men in de praktijk vaak aangewezen op een proceduregerichte taal.

Nu kan het probleem van de haringverkoper ook heel goed in een proceduregerichte taal, met gebruik van random generatoren voor een Poisson-verdeling en een uniforme verdeling, worden geprogrammeerd.

Ter illustratie volgt hier een eenvoudige versie van een computerprogramma van de haringkar met één bediende in pseudocode.

Dit programma is relatief eenvoudig om te zetten in BASIC, Pascal, enzovoorts op de eigen microcomputer of de (grote) computer waarover u kunt beschikken.

```
gegevensstructuur:
array wachtrij 1:20
laatst: een pointer t.b.v. de wachtrij.

     programma haringkar
begin initialiseer;
      leesin aankomstgemiddelde, simulatietijd, A, B;
      while  aankomsttijd < simulatietijd do
             aantalklanten := aantalklanten + 1;
             laatst := laatst + 1;
             aankomsttijd := aankomsttijd + negexp(1/aankomstgemiddelde);
             wachtrij[laatst] := aankomsttijd;
             while  verkopersklok ≤ aankomsttijd do
                    wachttijd := verkopersklok - wachtrij[1];
                    if  wachttijd < 0
                        then
                             wachttijd := 0
                             verkopersklok := wachtrij[1] + uniform(A,B)
                        else
                             verkopersklok := verkopersklok + uniform(A,B);
                    endif;
                    somwachttijd := somwachttijd + wachttijd;
                    for  n := 1 step 1 until laatste do
                         wachtrij[n] := wachtrij[n+1];
             enddo
      enddo
end  pseudoprogramma.
```

Commentaar

De variabele 'verkopersklok' bevat de tijd waarop de verkoper klaar is met het helpen van de klant.

De array 'wachtrij' bevat de aankomsttijden van de klanten die nog niet geholpen zijn.

Zodra een klant aankomt, gaat hij achter in de rij staan, ook al is de rij leeg.

Zodra de verkoper vrij is, helpt hij de eerste klant uit de rij.

De aankomsttijd houdt het verloop van de tijd bij als een soort 'winkelklok'.

De beide kansverdelingsfuncties 'negexp' en 'uniform' zullen als procedures moeten worden ontworpen in algemene programmeertalen. In Simula zijn ze standaard aanwezig. Er zal gewoonlijk nog een parameter nodig zijn die in de procedures de startwaarde van de gebruikte randomgenerator representeert.

Ad c. Na het schrijven van het programma moet dit worden getest op diverse soorten van fouten:
1. syntactische fouten
2. semantische fouten
3. logische fouten.

Als onderdeel van het vertaalprogramma (compiler) van de gebruikte hogere programmeertaal is altijd een programma aanwezig dat het programma controleert op taalfouten. Dit zijn fouten tegen de syntaxis van de betreffende taal. De *syntaxis* geeft de formele regels die worden gebruikt voor de definitie van de taal. In plaats van syntaxis spreekt men wel van de grammatica.

Terwijl in de syntaxis van een taal wordt vastgesteld welke symbolen gebruikt mogen worden en hoe deze symbolen aaneengeregen moeten worden tot woorden en tot zinnen, gaat het in de *semantiek* om de betekenis van de gebruikte constructies. Wanneer een programma derhalve op semantische fouten wordt gecontroleerd dan wordt nagegaan of in het programma wellicht constructies worden gebruikt waaraan door de compiler een andere betekenis wordt gegeven dan door de programmeur bedoeld wordt. Een eenvoudig voorbeeld hiervan is:

```
Y := X**3
```

In sommige programmeertalen wordt met '**' de rekenkundige bewerking machtsverheffen bedoeld. Als men nu in plaats van machtsverheffen de bewerking vermenigvuldigen op het oog heeft, dus bijvoorbeeld

```
Y := X*3
```

dan is er sprake van een semantische fout.

Dit soort fouten wordt door de syntax-controle niet ontdekt,

zodat hiervoor andere methoden gebruikt moeten worden. Te denken valt hierbij aan het 'draaien' van een programma met een beperkt aantal invoergegevens en van tevoren vast te stellen welke uitkomsten op grond van deze invoergegevens te verwachten zijn. Blijken nu de gerealiseerde uitkomsten af te wijken van de verwachte dan zijn één of meer semantische fouten mogelijk.

Van *logische fouten* is sprake als het programma niet overeenkomstig de bedoelingen van het model wordt geschreven. In andere woorden het 'programmamodel' is geen getrouwe afbeelding van het symbolische model. Voor de controle op logische fouten zijn diverse maatregelen mogelijk, zoals het geven van tussenuitvoer op verschillende plaatsen in het programma. Het programma controleren aan de hand van deze tussenuitvoer is eenvoudiger dan wanneer de eindresultaten aan de hand van de invoergegevens getoetst moeten worden.

Behalve dat voor deze testfase de hierboven bedoelde logische fouten opgespoord kunnen worden, is het ook mogelijk hieraan direct de modelverificatie te koppelen. Aan dit onderwerp besteden we in de volgende subparagraaf enige aandacht.

I.3.6 Modelverificatie

Alvorens nu de werkelijke simulatie in de vorm van het experimenteren met het model kan beginnen, zal nog vastgesteld moeten worden of het model in overeenstemming is met het beeld dat wij ons van het probleem hebben gevormd. Het model moet door de onderzoekers èn hun opdrachtgevers als een betrouwbare afbeelding geaccepteerd worden. Daarvoor moet het model aan een zogenaamde *plausibiliteitstoets* worden onderworpen. Deze test kan o.a. inhouden dat een bepaalde invoer wordt voorgelegd aan enkele deskundigen en dat hun mening wordt gevraagd over het gedrag van het systeem. Vervolgens worden deze resultaten vergeleken met de uitkomsten van het model. Deze vergelijking kan dan tot een verwerping of juist een acceptatie van het model leiden.

Deze plausibiliteitstest wordt gevolgd door het testen van de *consistentie* van het model. Een model wordt consistent genoemd als er geen tegenspraak uit afgeleid kan worden. Het onderzoek op consistentie vindt geheel plaats binnen het model. Men kan hierbij gebruik maken van zogenaamde waarheidstafels.

Hoewel de modelverificatie op het eerste gezicht vrij eenvoudig lijkt, is zij dat in de praktijk beslist niet. Dit komt mede door de volgende vraagstukken die zich bij een simulatie-onderzoek voordoen:

a. Geven de random generatoren toevalscijfers die voldoen aan de daaraan te stellen eisen? (Zie hoofdstuk III.)

b. Welke startwaarden moeten we invullen alvorens met het simuleren te beginnen?

c. Wat is de juiste steekproefomvang en hoeveel steekproeven zijn noodzakelijk?

De beide laatste vragen komen meer uitgebreid aan bod in hoofdstuk IV. In de volgende subparagraaf lopen we hierop in het kort vooruit.

I.3.7 Keuze van de proefopzet ('experimental design')

Hoe ontwerpen we nu een experiment dat de verlangde informatie gaat opleveren? Hierbij is een bekend spreekwoord van toepassing:

"Bezint, eer ge begint."

Ga, met andere woorden, niet in het wilde weg experimenteren, maar ontwerp een plan om zodoende de kosten van het onderzoek zo laag mogelijk te houden ten gunste van zoveel mogelijk informatie.

Wanneer we het simulatiemodel beschouwen als een 'black-box'-model, dan zijn er twee groepen van variabelen, namelijk de invoer- en de uitvoervariabelen interessant. In de literatuur over het onderwerp 'experimental design' worden deze groepen ook wel aangeduid als 'factoren' en 'responses'. Het gaat er in dit stadium van het onderzoek om vast te stellen:
a. welke uitvoervariabelen willen we meten;
b. met welke invoervariabelen gaan we experimenteren?

In ons eenvoudige model kiezen we bijvoorbeeld als uitvoervariabelen de gemiddelde wachttijd en de gemiddelde lengte van de wachtrij. De variabelen die hierop invloed uitoefenen, zijn de tussenaankomsttijd van de klanten en de bedieningstijd. Beide variabelen zijn stochastisch en worden beschreven door kansverdelingen met hun parameters.

Hoewel in het model het aantal variabelen klein is evenals het aantal parameters, is het aantal mogelijke experimenten eigenlijk al onbeperkt. Op grond van economische overwegingen zullen we ons dus beperkingen moeten opleggen. Deze beperkingen zouden in dit geval hierin kunnen liggen, dat we aannemen dat met de verdeling van de tussenaankomsttijd en de parameters niet geëxperimenteerd wordt. Dus, zoals in veronderstelling 1 in § I.3.3:

$$f(x) = \lambda e^{-\lambda x}$$

maar nu met: $\lambda = 2$ *).

Behalve deze beslissing moet ook een beslissing worden genomen omtrent de grootheden A en B van de uniforme verdeling. Het gaat erom goed afgestemde keuzen te maken van de waarden A en B.

*) De negatief exponentiële verdeling heeft slechts één parameter, namelijk $\mu = \sigma = 1/\lambda$.

> Men dient de keuze van de factoren zo in te richten, dat hierdoor het aantal experimenten kan worden beperkt en aan de opdrachtgever toch betrouwbare informatie kan worden verschaft.

Een mogelijkheid zou kunnen zijn één experiment uit te voeren met de experimenteel gevonden waarden van A en B, één experiment met waarden die verwacht worden bij twee verkopers en vervolgens een derde experiment met waarden van A en B tussen de eerste. Op grond van de uitkomsten van de uitvoervariabelen kan dan worden onderzocht welk soort verband (bijvoorbeeld lineair) er bestaat tussen de invoervariabelen en de uitvoervariabelen.

Elk experiment dat we doen vraagt om een doorrekening van 'computerruns' of zonder meer van 'runs'. Het aantal 'runs' kan, als we alle mogelijke experimenten willen onderzoeken, in een eenvoudig model reeds zeer groot worden. Als er in een model vier factoren zijn en elke factor heeft drie verschillende waarden dan zijn totaal $3^4 = 81$ combinaties mogelijk. Het is daarom van groot belang het aantal runs zoveel mogelijk te beperken.



I.3.8 Analyse van de resultaten

Als na het succesvol aflopen van de vorige fasen het model tenslotte wordt doorgerekend door de computer, krijgen we de gewenste uitvoer in de door ons gekozen vorm. We zijn dan in staat om aan de hand van deze uitvoer bepaalde conclusies te trekken.

Eerst zal bekeken moeten worden of het model vergelijkbaar is met de werkelijkheid. Dit betekent dat bij het model van de haringkar met één bedienend persoon de wachttijden, rijlengte, enzovoorts overeen moeten komen met de ervaring van de haringverkoper.

Wanneer deze vergelijking (gedaan m.b.v. statistische toetsen) goed wordt doorstaan, noemen we het model 'gevalideerd'. Daarna kan het model worden gebruikt om de verschillende alternatieven

door te rekenen en te evalueren. We bekijken dus de invloed die het aantrekken van een bediende heeft op de wachttijden. Omdat het model ons op grond van de resultaten voldoende vertrouwen inboezemt, zijn we bereid om aan de resultaten van het model, waarbij het (nog) niet bestaande systeem wordt onderzocht, voldoende waarde te hechten om de besluitvorming (wel of niet een bediende erbij) te ondersteunen. Verdere aspecten van validatie en betrouwbaarheid komen later in dit boek aan de orde.

I.4 OPGAVEN

Opgave 1

Vroeger, vóór de invoering van automatisering was de inschrijving van studenten aan de Technische Hogeschool te Delft een jaarlijks terugkerend evenement.
De aankomende studenten werden langs een schier eindeloze rij tafeltjes geleid, die zo strategisch stonden opgesteld, dat men geen enkel tafeltje kon overslaan. Bij elk tafeltje moest/kon/mocht er iets gebeuren.
Enkele voorbeelden:

0. Het 'doorlichten' t.b.v. het T.B.-onderzoek. Aan het maken van een röntgenfoto viel slechts bij hoge uitzondering te ontkomen.
1. Ruim 90% van de studenten kreeg een 'prik'. (De bij het T.B.-onderzoek behorende Mantoux-reactie.)
2. Uitreiken van collegekaart op vertoon van stamkaart.
3. Krijgen studierooster.
4. Kopen studiegids.
5. Medische verklaring.
6. Studiebeurs/toelage.
7. Studie-vereniging(en).
8. (Vele) studentenverenigingen.
9. Sportverenigingen.
10. Culturele lidmaatschappen, enzovoorts, enzovoorts.

Aan sommige tafeltjes zaten 2 of meer 'bedienden'. Toch kon men voor de gehele gebeurtenis wel enige uurtjes uittrekken. Vooral het doorlichten was een bottle-neck, zodat zich voor de ingang een lange wachtrij vormde.

Het beschreven systeem laat zich met behulp van simulatie geschikt onderzoeken.

1. Stel een zo gedetailleerd mogelijke doelstelling voor het onderzoek op.
2. Construeer van het systeem een schematisch model (m.b.v. een schets).
3. Definieer en klassificeer de relaties, variabelen en parameters.
4. Welke verdelingsfuncties kunt u definiëren?
5. Welke invoergegevens moet men aan het reële systeem ontlenen?

Opgave 2

De vraag per dag naar een artikel is gegeven door een Poissonverdeling met gemiddelde 2. Zie de volgende tabel.

Vraag	Kans
0	0.135
1	0.271
2	0.271
3	0.180
4	0.090
5	0.036
6	0.012
7	0.004
8	0.001

a) Simuleer 20 dagen en druk de vraaghoeveelheid per dag af in een frequentietabel. Dit kan gemakkelijk met 'handsimulatie'. Hoe richt u het experiment dan in?

b) De in a) geproduceerde reeks wordt gezien als een resultaat, waarvan men zegt, dat men hieruit zou kunnen afleiden dat een Poissonverdeling past.
Toets deze uitspraak m.b.v. de χ^2-goodness of fit test.
Neem per klassegrootte minimaal 3 (afgerond).
(Neem voor rekenwerk λ geheel, $e = 2,7$, $e^2 = 7,3$). *)

c) Wat verwachtte u als resultaat, nog voordat u de toets uitvoerde?

d) Wat toetst u in feite? (*Niet* de uitspraak in b).)

Opgave 3

Een van de toepassingen van een informatiesysteem, is dat men telefonisch een verzoek om inlichtingen, bepaalde benodigde gegevens, enzovoorts kan doen.
In een bepaalde concrete situatie duurt een telefonisch verzoek om informatie gemiddeld 2 minuten. De inkomende verzoeken blijken 'Poisson-verdeeld' te zijn.
De volgende situatie is: C belt op, A en B wachten op hun beurt voor hetzelfde toestel, zie figuur:



C belt reeds één minuut op zeker moment, dat A net arriveert. Hoe lang verwacht A te moeten wachten, alvorens zijn verzoek te kunnen plegen, gerekend vanaf het moment waarop hij aankomt? Ontwerp een simulatieprogramma om dit probleem op te lossen en controleer het antwoord.

Opgave 4

In een detailhandel blijkt, dat de verkopen van een van de produkten in het assortiment een Poissonverdeling volgen. De tabel geeft hierover nadere informatie.
Op het eind van elke 5-daagse werkweek wordt de winkelvoorraad beschouwd en wanneer er minder dan 6 exemplaren van het artikel aanwezig zijn, wordt een bestelling geplaatst voor 10 eenheden. Elke order wordt 3 of 4 werkdagen (met gelijke kans) na de bestelling geleverd.
U wordt gevraagd dit systeem te simuleren voor een tijd van 5 werkweken. Begin de simulatie aan het begin van een week, terwijl op dat moment de voorraad 9 stuks bedraagt en er geen uitstaande order is.
Aan het einde van elke dag moet het voorraadniveau uitgevoerd worden.
We zijn ook geïnteresseerd in het aantal dagen, waarop het artikel is uitverkocht en het aantal artikelen dat men gedurende die 5 weken om die reden niet heeft kunnen verkopen.

---

*) Een voorbeeld van de gevraagde toets is gegeven in Appendix A.

Tabel:

de vraag per dag	waarschijnlijkheid
0	0.30
1	0.36
2	0.22
3	0.09
4	0.03

1. Voer een handsimulatie uit.
2. Maak een formele programmabeschrijving, hetzij in programmastructuurdiagram (PSD), hetzij in een gestructureerde pseudocode (M-BASIC, Pascal of SIMULA).
3. Indien mogelijk: geef een computeruitvoer.

Opgave 5

1. *Probleembeschrijving*

Een taxichauffeur in een grote stad, die verdeeld is in drie wijken: A, B en C, heeft om aan passagiers te komen in de wijken A en C drie mogelijkheden en in wijk B twee mogelijkheden. De verschillende alternatieven zijn:
1. hij kan de wijk waarin hij zich bevindt op goed geluk doorrijden totdat hij een passagier vindt;
2. hij kan naar een taxistandplaats gaan en daar wachten op een passagier;
3. hij kan de wagen op een willekeurige plaats langs de kant zetten en daar wachten op een passagier, dan wel op een radio-oproep.

In wijk B bestaat het derde alternatief niet. Voor de taxichauffeur is het interessant te weten welke strategie, d.w.z. welke combinatie van alternatieven, het beste door hem gekozen kan worden teneinde een zo hoog mogelijke opbrengst te verkrijgen.

2. *Noodzakelijke gegevens*

Om de optimale strategie te kunnen berekenen, moeten we over kwantitatieve gegevens beschikken. Allereerst moeten we de kansen om een passagier te krijgen voor elk van de drie c.q. twee alternatieven in elk van de drie mogelijke wijken weten. Verder moeten we informatie hebben omtrent de (gemiddelde) opbrengst per rit voor elk van de gevallen. We nemen aan, dat de taxichauffeur ons deze cijfers op grond van zijn jarenlange ervaring kan geven; ze zijn vermeld in de volgende tabel.

Toestand i	Alternatief k	Kans A	B	C	Gemiddelde opbrengst per rit in guldens A	B	C
A	1	1/2	1/4	1/4	10	4	8
	2	1/16	3/4	3/16	8	2	4
	3	1/4	1/8	5/8	4	6	4
B	1	1/2	0	1/2	14	0	18
	2	1/16	7/8	1/16	8	16	8
C	1	1/4	1/4	1/2	10	2	8
	2	1/8	3/4	1/8	6	4	2
	3	3/4	1/16	3/16	4	0	8

Het zich bevinden in een bepaalde wijk wordt aangeduid als 'toestand'. De eerste matrix geeft voor elk alternatief de kansen op een passagier voor wijk A, wijk B, respectievelijk wijk C. Zo is in toestand A de kans op een rit naar B bij het tweede alternatief 3/4. De tweede matrix geeft de gemiddelde uitkomsten van elk soort rit voor de verschillende alternatieven.

Vragen

1. Formuleer aan de hand van de probleemformulering en de gegevens een model.
   Ga na welke variabelen en factoren een rol zouden kunnen spelen, en welke men in het model opneemt.
   Definieer de doelstelling van de simulatie.
2. Probeer het model 'na te spelen' en formuleer aan de hand hiervan het proces.
3. Analyseer het probleem en stel de beste strategie van de taxichauffeur vast. Motiveer uw antwoord.
4. Schrijf een programma in 'geformaliseerd Nederlands'.
5. Geef enige overwegingen bij het experimenteel ontwerp.
6. Schrijf het programma in een programmeertaal naar eigen keuze.
7. Beschouw de resultaten die door de computer worden uitgevoerd en geef uw bevindingen omtrent de beste strategie.

HOOFDSTUK II

Simulatie: Uitgangspunten en Bouwstenen

II.1 INLEIDING

In dit hoofdstuk zullen we eerst opnieuw ingaan op de vraag: 'Wat is simulatie?'. Verschillende opvattingen zullen (kort) aan de orde komen. De opvattingen hebben gemeen, dat de achtergrond om te simuleren een informatiebehoefte is. Men wil iets weten over een bepaald systeem, bijvoorbeeld het voorraad- of het personeelsverloop. Om aan deze informatiebehoefte te kunnen voldoen, is een model noodzakelijk. Het model is een essentiële bouwsteen van een simulatie-onderzoek. We zullen daarom vrij veel aandacht besteden aan het modelbegrip. In § 4 gaan we in op het model als afbeelding, terwijl in § 5 een aantal modelindelingen wordt behandeld. Termen als dynamisch, stochastisch, discreet en continu worden in deze paragraaf toegelicht.

In een model wordt voor de beschrijving van de werkelijkheid gebruik gemaakt van variabelen, van relaties tussen de variabelen en van parameters. In § 6 geven we een toelichting op de verschillende soorten variabelen, relaties en parameters die in simulatiemodellen gebruikt (kunnen) worden. In § 7 wordt nader ingegaan op doel en gebruik van simulatiemodellen. De aan een simulatiemodel te stellen eisen zullen namelijk afhankelijk zijn van het doel van het onderzoek. Ook dit hoofdstuk wordt afgesloten in § 8 met een aantal opgaven.

II.2 SIMULATIE: PLAATSBEPALING EN DEFINIERING

In § I.2 is reeds ingegaan op simulatie als methode van onderzoek. We hebben daar een globale beschrijving gegeven van de fasen van een simulatie-onderzoek. Een definitie hebben we achterwege gelaten. Het doel van deze paragraaf is onze visie op simulatie nader uiteen te zetten. We zullen ook ingaan op een tweetal andere opvattingen.

De term simulatie is afgeleid van het Latijnse werkwoord 'simulare' hetgeen o.a. nabootsen betekent. Zo worden in de natuurwetenschappen met behulp van windtunnels uiteenlopende luchtstromingen en het effect daarvan op diverse voorwerpen nagebootst. In het laboratorium voor Fysische Technologie van de T.H. Delft wordt met behulp van simulatie onderzoek verricht naar de oorzaken van algengroei in drinkwaterbekkens. En in de gedragswetenschappen tracht men een beter inzicht in het menselijk gedrag te krijgen door met behulp van proefpersonen bepaalde concrete situaties na te bootsen in een rollenspel.

Het belangrijkste argument voor het gebruik van simulatie in veel situaties is, dat er geen analytische technieken beschikbaar zijn voor het vinden van een oplossing of dat deze technieken te duur zijn. Simulatie wordt dan als een *laatste redmiddel* ('a method of last resort') beschouwd. Vooral beoefenaren van de operations research geven de voorkeur aan analytische technieken boven heuristische*) technieken, zoals simulatie.

In de natuurwetenschappelijke omgeving wordt simulatie vooral gezien als een mogelijkheid tot het *uitvoeren van experimenten.* Simulatie is een methodiek waarin geëxperimenteerd wordt met modellen van complexe processen. Door middel van deze experimenten wordt de kennis van de processen vergroot, zodat hiervan bij de bouw van een werkelijk systeem gebruik gemaakt kan worden. Een mooi voorbeeld zijn de Delta-werken in Zeeland.

In de laatste jaren komt simulatie steeds meer naar voren als een zelfstandige *methode van onderzoek*, vooral ten behoeve van vraagstukken in de sociale wetenschappen. Met een zelfstandige methode van onderzoek wordt bedoeld, dat simulatie kan bijdragen tot verbetering van de modellen, tot de analyse van beslissingsprocessen en daardoor tot verbetering van theorievorming. Simulatie biedt de mogelijkheid beslissingsprocessen van een inhoud te voorzien. Deze processen, die in de operations research en de econometrie vaak als een 'black box' worden beschouwd, kunnen met simulatie worden beschreven en geanalyseerd.

Ook wij beschouwen simulatie als een methode van onderzoek. Een in dit verband passende definitie van deze onderzoekmethode vinden we bij Shannon. **)

---

*) De term heuristische methode staat voor: 'vindmethode', 'vuistregels' in tegenstelling tot de term analytisch, waarbij men een mathematisch algoritme formuleert.

**) Zie R.E. Shannon, Systems Simulation: the art and science, New York, 1975, pag. 2.

"Simulation is the process of designing a model of a real system, and conducting experiments with this model for the purpose either of understanding the behaviour of the system *or* of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system."

Het woordje 'or' moet in deze definitie niet verkeerd worden geïnterpreteerd. Het woordje zou de indruk kunnen wekken, dat er een tegenstelling bestaat tussen 'verklaren' en 'voorschrijven'. Wij zijn, evenals Bosman*), van mening dat er geen tegenstelling bestaat, maar dat om te kunnen voorschrijven eerst verklaren een noodzaak is.

Shannon voegt aan deze definitie ter verdere verduidelijking nog het volgende toe:

Simulatie is een experimentele en toegepaste methodologie waarvan het doel is:
1. het gedrag van systemen (vooral het gedrag in de tijd) te beschrijven;
2. theorieën of hypothesen te construeren die het waargenomen gedrag verklaren;
3. deze theorieën te gebruiken om het gedrag in de toekomst te voorspellen, dat wil zeggen de effecten te beschrijven die zullen optreden door veranderingen of wijzigingen in het systeem of in de wijze van de werking van het systeem (vert. auteurs).

Hoewel in de veelheid van definities die van Shannon ons het meest aanspreekt, blijft de vraag of de gegeven definitie wel discriminerend genoeg is voor het begrip simulatie. Als we in de definitie bijvoorbeeld het woord 'Simulatie' vervangen door 'Operations Research', klopt de definitie dan ook? Het antwoord zou bevestigend kunnen zijn en dat is de bedoeling niet.

In onze opvatting is simulatie meer dan een techniek, zoals bijvoorbeeld lineaire programmering. Simulatie is een methode van onderzoek die in belangrijke mate geschikt is voor het onderzoek van sociale systemen. Simulatie is een specifieke verzameling van technieken, waarbij het gemeenschappelijke moet worden gezocht in de wijze waarop men problemen wenst op te lossen. Dit gemeenschappelijke wordt ontleend aan de modelcyclus (zie hoofdstuk III).

Een belangrijk verschil tussen Simulatie en Operations Research ligt ook hierin, dat het te ontwerpen model bij OR geformuleerd moet zijn in de vorm van wiskundige relaties: het model is een algorithmische beschrijving van de werkelijkheid. Wanneer men het

---

*) A. Bosman, Een metatheorie over het gedrag van organisaties, Leiden, 1977, pag. 175.

OR-model doorrekent, komt er één en slechts één antwoord uit: het optimale antwoord. Bij simulatie behoeft men het model niet in wiskundige relaties te formuleren: men bootst het systeem na in de vorm van het proces of een reeks gebeurtenissen. Dit levert natuurlijk meer mogelijkheden in die gevallen waarin het wiskundige formuleren moeilijk of onmogelijk is. Aan de andere kant is het antwoord niet eenduidig en daardoor soms onderhevig aan twijfels omtrent de betrouwbaarheid.

Concluderend kan worden gesteld, dat men met simulatie modellen ontwerpt en onderzoekt, niet alleen met het oog op de oplossing van organisatiekundige, of meer in het algemeen van sociale problemen, maar ook met het oog op de verdieping van theoretisch inzicht op zichzelf, als tussenstap tot verklaringen, omdat 'experimentatie' in de werkelijkheid sterk aan banden is gelegd.

In de volgende paragraaf zullen we met name nader ingaan op de argumenten vóór, en ook tegen, het gebruik van simulatie als onderzoekmethode.

II.3 WANNEER (GEEN) SIMULATIE?

Door Van Peursen en Boendermaker*) worden drie redenen genoemd die tot simulatie-onderzoek aanleiding kunnen geven.

a. Het uitvoeren van experimenten in de werkelijkheid kan technisch (vrijwel) onmogelijk zijn om redenen die uit de aard van het onderzoeksobject zelf voortvloeien. Dit doet zich voor, wanneer het bestuderen van een proces zoveel tijd kost, dat het tijdens de proefneming reeds aanzienlijk zou veranderen. Ook is dit het geval, wanneer men op korte termijn een economisch beleid moet ontwerpen, terwijl het hiervoor benodigde historische feitenmateriaal niet meer volledig te achterhalen valt.
b. Het experiment kan om economische redenen niet uitvoerbaar zijn. Zo is het bestuderen van alternatieve wegtracés door deze werkelijk aan te leggen, te kostbaar, evenals het bouwen van een aantal proefopstellingen om op ware schaal alternatieve condities voor een chemisch produktieproces uit te proberen.
c. Proefnemingen in de werkelijkheid kunnen tot consequenties leiden die in ethisch opzicht onaanvaardbaar zijn. Hierbij kan men denken aan onderzoekingen naar de verspreidingsmogelijkheid van besmettelijke ziekten in de epidemiologie of het testen in de praktijk van bepaalde polemologische modellen, zoals die voor de bewapeningswedloop tussen landen.

Men experimenteert dus met modellen van processen, omdat deze laatste niet voor directe proefneming toegankelijk zijn en/of omdat de complexiteit van de diverse variabelen zo groot is dat hierin inzicht moet worden verkregen alvorens tot verklaringen en/of oplossingen van de gestelde problematiek kan worden gekomen.

De genoemde drie redenen richten zich in hun argumentatie sterk op de mogelijkheid van het doen van experimenten. Het genereren en doorrekenen van alternatieven, met name wanneer experimenteren in de realiteit problemen met zich brengt wordt als belangrijkste motief naar voren gehaald. Hiernaast kunnen andere, niet minder belangrijke overwegingen als argument vóór het gebruik van simulatie worden genoemd.

d. Voor theorievormend onderzoek kan simulatie een belangrijke rol vervullen. We hebben hierop in de vorige paragraaf reeds gewezen. Met name de mogelijkheid beslissingsprocessen van een inhoud te voorzien, is zeer belangrijk.
e. Simulatiemodellen zijn met betrekking tot de te gebruiken wiskundige technieken in het algemeen niet lastig. De modellen zijn vaak overzichtelijk en lenen zich daardoor goed voor

---

*) C.A. van Peursen en S. Boendermaker, in: Intermediair, jg. 12, nr. 3, 1976, pag. 25.

instructiedoeleinden. De implementatie van een simulatiemodel kan zodoende gemakkelijker verlopen dan van bijvoorbeeld complexe wiskundige modellen. Vanzelfsprekend is in dit verband veel afhankelijk van de mensen die betrokken zijn bij het implementatieproces.

f. Tenslotte willen we als argument voor simulatie noemen de mogelijkheid om de gewenste gegevens te genereren. In veel probleemsituaties ontbreekt het de onderzoeker aan basisgegevens. De administratie is vaak financieel georiënteerd en is niet in staat gegevens omtrent wachttijden, omsteltijden e.d. te leveren. Vaak worden in organisaties alleen de daadwerkelijke beslissingen vastgelegd, terwijl de gegevens om hiertoe te komen worden 'vergeten'. Een essentieel onderdeel van simulatie zijn de zogenaamde Monte Carlo-procedures, waarmee het mogelijk is gegevens beschikbaar te stellen zonder dat deze in de organisatie aanwezig behoeven te zijn.

Hoewel er dus vele redenen zijn die pleiten vóór het gebruik van simulatie, zijn er vanzelfsprekend ook situaties dat naar andere middelen moet worden omgezien.

1. Het ontwikkelen van een simulatiemodel is vaak tijdverslindend en bij intensief computergebruik kostbaar. Hoewel door een goed experimenteel ontwerp (zie hoofdstuk IV) de kosten beperkt kunnen worden, kunnen deze in bepaalde gevallen simulatie als onderzoekmethode uitsluiten.

2. Simulatie kan in bepaalde gevallen de schijn wekken een systeem betrouwbaar weer te geven. Er doen zich tijdens een onderzoek technische problemen voor die door deskundigen opgelost dienen te worden. Als zo'n deskundige niet beschikbaar is, kan beter worden afgezien van simulatie.
Zo kunnen bijvoorbeeld zaken als runlengte en startwaarden bij een onjuiste keuze tot op het oog betrouwbare, maar in werkelijkheid tot onbetrouwbare resultaten leiden.

3. Zoals reeds eerder opgemerkt levert simulatie geen optimale oplossingen. Of beter gezegd, in de meeste gevallen weten we niet of een oplossing optimaal is.
Nu geldt voor veel - vooral slecht gestructureerde - problemen, dat het zoeken naar de optimale oplossing een onmogelijke zaak is, onder meer door het feit dat er zeer vele alternatieven bestaan. Bovendien kennen we niet alle alternatieven. In zo'n geval zoekt men naar een 'bevredigende' oplossing, een oplossing, die een bepaald aspiratieniveau bevredigt.
Toch zal - afhankelijk van het probleemgebied - het voor sommigen niet aanvaardbaar zijn, dat men van een bepaalde probleemoplossing niet kan zeggen, dat deze de optimale oplossing is.
Als opdrachtgevers om deze reden geen vertrouwen in simulatie hebben, zal beter naar een andere oplossingsmethode kunnen worden gezocht.

II.4 HET MODELBEGRIP

Met behulp van simulatie zal men dus een systeem willen nabootsen. Dit nabootsingsproces heeft tot doel inzicht te verkrijgen in de werking van dit systeem om zodoende voorspellingen te kunnen doen.

Als we het hier hebben over een systeem, dan bedoelen we een stukje van de waarneembare werkelijkheid, bezien door de bril van de onderzoeker. Alvorens we dit stukje werkelijkheid, ook wel aangeduid als het objectsysteem kunnen gaan onderzoeken, zullen we er eerst een beschrijving van moeten geven. Zo'n beschrijving van de werkelijkheid wordt veelal aangeduid als afbeelding of model.

> Een *model* definiëren we als *een afbeelding van een objectsysteem,* met als doel inzicht te verkrijgen in het gedrag van dat systeem.

In deze definitie verstaan we onder het gedrag een specificatie van de beslissing die voortvloeit uit het beslissingsproces. Dit proces wordt ook wel 'handelen' genoemd.

Met andere woorden: met een model wordt geprobeerd een zo getrouw mogelijke weergave te geven van een beslissingsproces. Een model is een beschrijving met als eerste doel meer inzicht te krijgen in de probleemsituatie om vervolgens naar mogelijke oplossingen te kunnen zoeken.

We kunnen dit ook nog toelichten aan de hand van een eenvoudig voorbeeld. Het geld dat een gezin iedere maand reserveert voor de huishouding kan gebaseerd zijn op de regel:

$$\hat{B}(t) = B(t-1) + B(t-1)*k/100$$

$\hat{B}(t)$ = gereserveerde huishoudgeld in maand t (gereserveerd aan het begin van de maand)
$B(t-1)$ = werkelijk benodigde huishoudgeld in maand t-1
k = procentuele verandering in kosten van levensonderhoud in de voorgaande periode volgens opgave CBS.

In dit voorbeeld wordt het handelen van het gezin weergegeven door de bovenstaande beslissingsregel. Het gedrag dat uit het handelen volgt, is de reservering van meer of minder huishoudgeld.

We staan ook nog even stil bij de omschrijving 'een afbeelding van een objectsysteem'. Hiermee bedoelen we, dat een model een weergave is van een stukje realiteit, ook wel aangeduid als een reëel systeem. Zo'n weergave is meestal zeer beperkt. Een model bevat in het algemeen maar een deel van de facetten van het reële systeem. De bovenstaande beslissingsregel voor de bepaling van

het huishoudgeld is maar heel beperkt. Hij houdt bijvoorbeeld geen rekening met de inkomsten van het gezin of met onverwachte uitgaven. Als uitgangspunt voor het totale bestedingsproces kan de regel evenwel beslissingsondersteunend goede diensten bewijzen.

Om met een model inzicht te krijgen in een objectsysteem zal een model aan een aantal voorwaarden moeten voldoen. Deze voorwaarden kunnen samengevat worden als:

> Het model moet in structuur overeenkomen met het objectsysteem.

Wat houdt deze uitspraak in? Voor de toelichting keren we nog even terug naar het voorbeeld van de haringkar. Als meerdere mensen tegelijk geholpen willen worden, ontstaat een wachtrij. Deze rij kan gezien worden als een ordeningselement. We nemen in het model aan, dat alle klanten zich houden aan de discipline volgens de regel 'first in, first out' (fifo). In de realiteit van het objectsysteem zal heus wel eens van deze regel worden afgeweken. Mensen met haast krijgen of nemen voorrang. Ook zal de haringman misschien wel eens twee mensen tegelijk bedienen. In ons model houden we hiermee geen rekening en dat is, gegeven de doelstelling van het onderzoek ook niet noodzakelijk. Om inzicht te krijgen in de wachttijd van klanten en daardoor in de vraag, of meer bediening nodig is, komt het model voldoende overeen met de structuur van het objectsysteem.

Concluderend kunnen we dus vaststellen, dat de ordening van de elementen van het objectsysteem (aan te duiden als 'structuur') in het model zodanig moet zijn aangebracht, dat het doel van het onderzoek niet in gevaar komt. Door de complexiteit van de meeste objectsystemen is volkomen structurele identiteit van model en objectsysteem ook niet mogelijk. Steeds zal het doel van het onderzoek het model bepalen.

II.5 CLASSIFICATIE VAN MODELLEN

In de literatuur worden verschillende modelindelingen gegeven. Voor het onderwerp simulatie zijn vooral de volgende indelingen van belang:

1. statisch versus dynamisch
2. deterministisch versus stochastisch
3. discreet versus continu
4. iconisch en analoog.

We zullen op elk van deze indelingen kort ingaan.

1. Er bestaat in de literatuur geen eenstemmigheid over de betekenis van de begrippen *statisch* en *dynamisch*. In de systeemtheorie definieert men een systeem als statisch indien de in- en uitvoervariabelen gedurende een bepaalde periode T constant zijn. Veranderen de in- en uitvoervariabelen gedurende de periode T dan noemt men het systeem dynamisch. In de economie daarentegen spreekt men dan van 'stationaire' en van 'veranderlijke' toestand. Van een stationaire toestand spreekt men, indien de data (dat zijn de invoervariabelen) onveranderlijk blijven; wijzigen zij zich wel, dan heeft men te doen met een veranderlijke toestand.

De termen 'dynamisch' en 'statisch' worden in de economie gebruikt in relatie met de analyse van het verloop van de economische grootheden in de tijd. Als verondersteld wordt dat in een proces alle veranderingen van de economische grootheden zich onmiddellijk voltrekken, dan spreekt men van een statische analyse of van statica. In een dynamische analyse daarentegen bestudeert men het verloop van de economische grootheden in de tijd.

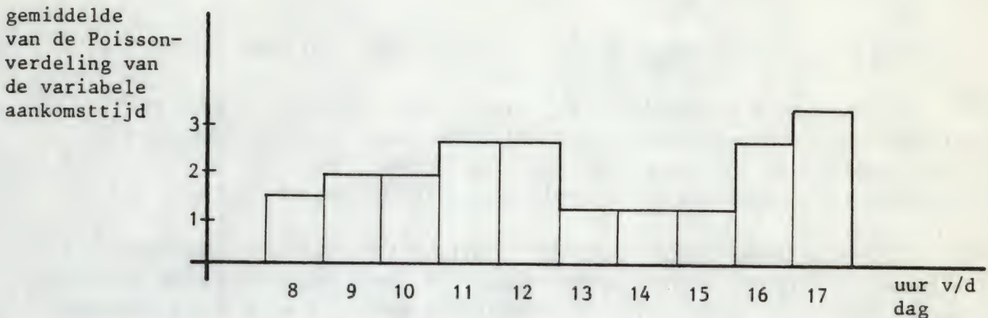
Beide betekenissen van de begrippen statisch en dynamisch zijn voor het onderwerp 'simulatie' van essentieel belang. Enerzijds zullen we in een simulatiemodel de invloed van veranderingen van de invoervariabelen op de uitvoervariabelen willen onderzoeken en anderzijds zullen we ook het verloop van de uitvoervariabelen in de loop van de tijd willen kennen.

De termen statisch en dynamisch moeten dus worden geïnterpreteerd afhankelijk van het zelfstandig naamwoord waaraan ze zijn toegevoegd. Een koppeling met het woord model leidt tot de omschrijving, zoals die in de systeemtheorie wordt gegeven; bij de relatie met het woord analyse gaat het om het tijdpad van de variabelen. De factor 'tijd' kan hierbij niet gemist worden als verklarende (onafhankelijke) variabele.

Wanneer we het model van de haringkar beschouwen, dan is dit in de gegeven vorm een statisch model. Immers de reeks waarden van de invoervariabelen 'aankomsttijdstip' en 'bedieningstijdstip' is niet afhankelijk van het moment waarop het systeem in onderzoek wordt genomen. M.a.w. het maakt voor de haringkar niet uit of we

het begin van de simulatie op maandagmorgen 10 uur stellen dan wel op vrijdagmiddag 5 uur.

Wanneer we echter de variabele 'aankomsttijd' afhankelijk doen zijn van het uur van de dag (zie figuur II.5a) en eventueel ook van de dag in de week dan wordt het systeem een dynamisch systeem; de variabele 'tijd' speelt dan een rol in het systeemgedrag.



Figuur II.5a

De uitvoervariabele 'wachttijd' zal nu niet alleen een verloop in de tijd volgen, dat afhankelijk is van de drukte in het systeem, maar een tijdreeks vertonen, waarin het uur van de dag als verklarende variabele een rol speelt.

Wiskundig kan men een dynamisch systeem eenvoudig formuleren. De uitvoervariabele y is een functie van de invoervariabele $x \rightarrow y = F(x)$.

Dan geldt voor een dynamisch systeem dat:

$$y_t = H_0 x_t + H_1 x_{t-1} + H_2 x_{t-2} + \dots\dots\dots\dots\dots$$

H is de invloed (de transformatiefunctie) van de invoerwaarden op iedere uitvoerwaarde y_t, waarbij de afstand in de tijd steeds groter wordt. De invloed H_p zal in het algemeen vanaf een bepaalde tijdsafstand p verwaarloosbaar klein worden.

De wachttijd die optreedt om 13 uur bij de haringkar zal sterker beïnvloed zijn door de waarden van de aankomst- en bedieningstijden tussen 12 en 13 uur dan door die tussen 11 en 12 uur, terwijl het effect van de vroege ochtenduren nihil zal zijn.

Wanneer de invoer/uitvoerrelatie kan worden geschreven als $y_t = H_0 x_t$, is y_t niet meer afhankelijk van de waarden van de invoer op eerdere tijdstippen en spreekt men van een statisch systeem.

2. Ook over de indeling deterministisch versus stochastisch bestaat geen eenstemmigheid. Wij verstaan onder deterministische modellen die modellen, waarin alle variabelen en de relaties tussen de variabelen 'vast' zijn, d.w.z. dat ze niet beïnvloed worden door waarschijnlijkheidsmechanismen.

Een voorbeeld van een deterministisch model is

$$Y(t) = 100 + 0,8X(t)$$

als $X(t)$ per periode één en niet meer dan één waarde heeft.

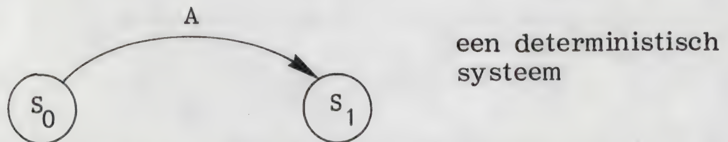
Een model is stochastisch als òf de variabelen òf de relatie(s) òf de variabelen èn de relaties stochastisch zijn, d.w.z. dat er 'toevalselementen' aanwezig zijn. Er bestaat een niet te voorspellen onzekerheid omtrent de waarde van variabelen of de toestand van een systeem.

Een andere naam voor stochastische variabele is kansvariabele. Als in de bovenstaande vergelijking X een stochastische variabele is en dus meerdere waarden kan aannemen, elk met een bepaalde kans, dan is het model stochastisch. We kennen deze kwalificatie eveneens toe als de vergelijking zou zijn:

$$Y(t) = 100 + 0,8X(t) + \varepsilon$$

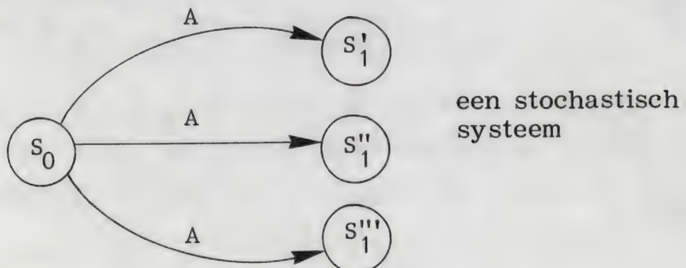
waarin $X(t)$ vast is en ε een kansvariabele met een bepaalde verdelingsfunctie is.

In het algemeen kan een deterministisch systeem als volgt worden voorgesteld:



Vóórdat aktiviteit A plaatsvindt, bevindt het systeem zich in toestand S_0. Na aktiviteit A is het systeem in toestand S_1 terecht gekomen.

Een stochastisch systeem daarentegen kan als volgt in beeld worden gebracht:



Na aktiviteit A kan het systeem zich - met 'n bepaalde waarschijnlijkheid - in één van de drie verschillende toestanden S_1 bevinden. Welke toestand dat zal zijn, is niet met zekerheid te voorspellen.

De haringkar is een voorbeeld van een stochastisch model. Het Du Pont model (bekend uit de bedrijfseconomie) dat een relatie weergeeft tussen het brutorendement (rentabiliteit van het totale vermogen: ROI) en een aantal financiële grootheden van een organisatie, is een voorbeeld van een deterministisch model.

3. *Continue* modellen verwijzen naar systemen waarbij de veranderingen in de toestand continu verlopen in de tijd. Als de veranderingen echter sprongsgewijs optreden, in opeenvolgingen van eindige tijdsperioden, dan spreken we over *discrete* systemen. De haringkar is een typisch voorbeeld van een discreet systeem.

Technische en fysische systemen zijn vaak continu, terwijl systemen uit economische en bedrijfskundige toepassingsgebieden veelal discreet zijn.

4. *Iconische modellen* zijn schaalmodellen, bijvoorbeeld een schaalmodel van een nieuw te ontwerpen Stadhuis of van een nieuw te ontwerpen auto.

Van *analoge modellen* spreken we wanneer bepaalde eigenschappen van grootheden in de werkelijkheid in het model worden weergegeven door analoge eigenschappen van modelgrootheden.

Een voorbeeld van een analoog model is een landkaart, waarmee een geografisch systeem afgebeeld wordt. Hierbij kunnen ook nog vaak bepaalde aspecten benadrukt worden, zoals bij een wegenkaart, een topografische kaart, een hoogtekaart, een kaart met klimatologische verschillen, enzovoorts.

Een ander voorbeeld is het gebruik van watermodellen met bakken, leidingen, kranen, enzovoorts, waarmee elektrische systemen kunnen worden nagebootst. Ook een bouwtekening die op schaal de maten van een te bouwen huis met de verschillende vertrekken en de plaats van zaken als lichtpunten, contactdozen, ramen, enzovoorts aangeeft, is een voorbeeld van een analoog model.

Opmerking 1

Omdat wij ons in dit boek beperken tot discrete computersimulatie zullen analoge systemen altijd omgezet worden naar discrete modellen. Zo zal een stelsel differentiaalvergelijkingen, dat in veel gevallen een analoog systeem blijkt te kunnen representeren, via een numerieke methode, in kleine stapjes rekenkundig gesimuleerd worden.

Opmerking 2

In deze paragraaf zijn in zekere mate de termen 'systeem' en 'model' door elkaar gebruikt.
Daar een model - een afbeelding van een systeem - zelf ook weer als een systeem kan worden opgevat, is het onderscheid niet altijd duidelijk of nodig. De classificatie van modellen gaat goeddeels ook op voor systemen.

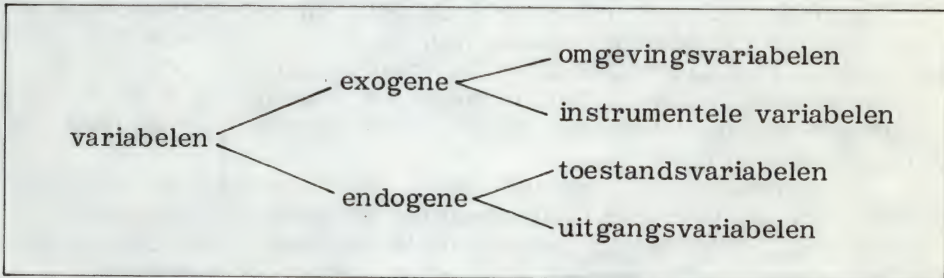
II.6 BESCHRIJVING VAN MODELLEN

Modellen, die wij gebruiken om te experimenteren, zijn systeembeschrijvingen die in een bepaalde vorm worden afgebeeld. We werken met symbolen om de grootheden in het objectsysteem weer te geven. Vaak ook zijn de relaties, die er tussen de grootheden bestaan in de vorm van stelsels vergelijkingen weergegeven. Modellen, die op deze wijze zijn opgebouwd, worden *symbolische modellen* genoemd.

Het type modellen waarop wij ons in dit boek concentreren kan worden gekenmerkt als symbolische modellen, die discrete stochastische processen afbeelden. Bij de beschrijving van objectsystemen met behulp van deze modellen maken we gebruik van variabelen en van parameters, terwijl de onderlinge afhankelijkheid door relaties wordt gepresenteerd. De drie begrippen: variabelen, relaties en parameters worden hieronder nader beschreven.

1. VARIABELEN

Variabelen kunnen in twee hoofdgroepen worden onderverdeeld:



Beide groepen variabelen worden op hun beurt weer verdeeld in twee verzamelingen. We bespreken deze variabelen nu per soort.

a. Exogene (of onafhankelijke) variabelen

Omgevingsvariabelen

Omgevingsvariabelen zijn variabelen die elementen buiten het te onderzoeken systeem symboliseren en die op dit systeem een zekere invloed uitoefenen zonder dat vanuit het systeem zelf hierop invloed kan worden uitgeoefend.

Zo is voor een onderneming de door de bank te berekenen debetrente in het algemeen een niet te beïnvloeden variabele en dus een omgevingsvariabele.

Instrumentele (of beslissings-) variabelen

Deze groep van variabelen is eveneens onafhankelijk, maar de beslisser kan, vaak binnen bepaalde grenzen de waarde ervan beïnvloeden. De beslisser kan ze gebruiken als instrument om er invloed mee uit te oefenen op de endogene of onafhankelijke variabelen.

Een voorbeeld hiervan is het belastingtarief, waarmee de overheid kan manipuleren om de consumptieve uitgaven in een volkshuishouding te beïnvloeden.

b. Endogene (of afhankelijke) variabelen

Toestandsvariabelen

Deze variabelen beschrijven de situatie van een systeem op een bepaald moment, bijvoorbeeld de voorraad, de gemiddelde levertijd, enzovoorts. "Ruwweg kan men stellen", schrijven Hanken en Reuver*), "dat in de toestandsvariabelen alle relevante historische informatie 'opgeborgen' ligt. De waarde van de toestandsvariabele op een bepaald moment, kan berekend worden uit de waarde van de toestandsvariabele op het voorgaande moment en de waarden van de onafhankelijke variabelen in de tijdsperiode tussen twee momenten."

Als illustratie geven we een voorbeeld van een debiteurenadministratie. De post debiteuren is de toestandsvariabele van dit subsysteem van de boekhouding. De grootte van deze post wordt enerzijds bepaald door de factuurbedragen, anderzijds door de betalingen van de debiteuren en de overboekingen naar de post dubieuze debiteuren.

'Factuurbedragen', 'betalingen' en 'overboekingen naar dubieuze debiteuren' zijn onafhankelijke variabelen. 'Factuurbedragen' en 'overboekingen naar dubieuze debiteuren' kunnen als beslissingsvariabelen worden beschouwd, terwijl de variabele 'betalingen' een ingangsvariabele is. Hoewel vanuit het systeem wel enige invloed op de betalingen uitgeoefend kan worden door bijvoorbeeld kortingen, aanmaningen, e.d. ligt de beslissing om te betalen uiteindelijk buiten het systeem.

De toestandsvariabele 'debiteuren' kan op elk moment worden vastgesteld als men kennis heeft van de hoogte van deze post op een voorgaand moment en van de waarden van de beslissingsvariabelen in de tijdsperiode tussen de beide momenten.

Uitgangsvariabelen

De uitgangsvariabelen zijn afhankelijke of endogene variabelen, die voortvloeien uit de probleemstelling. Voorbeelden zijn winst, rentabiliteit, energieverbruik, enzovoorts.

---

*) A.F.G. Hanken en H.A. Reuver, Inleiding tot de systeemleer, Leiden, 1976.

De vraag tot welke groep een bepaalde variabele behoort, hangt af van het onderzoek. Zo kan in ons voorbeeld de post 'debiteuren' behalve als toestandsvariabele ook als uitgangsvariabele worden gezien. Dit is afhankelijk van de probleemstelling. Het doel van een onderzoek kan bijvoorbeeld zijn na te gaan welke factoren van invloed zijn op de hoogte van de post 'debiteuren'. De variabele 'debiteuren' is dan zowel toestandsvariabele als uitgangsvariabele.

Een andere mogelijkheid is, dat een onderzoek wordt begonnen met als doel inzicht te krijgen in de kosten van het debiteurensysteem. De bij de debiteuren uitstaande gelden hebben een derving van inkomsten tot gevolg. Aan de afnemers wordt krediet gegeven. De kosten van dit krediet kunnen als een endogene variabele worden beschouwd, waarvan de waarde kan worden berekend uit de toestandsvariabele 'debiteuren' en een parameter 'kredietkosten per gulden per tijdseenheid'.

2. RELATIES

De verzameling relaties, welke tussen de elementen (variabelen) van een systeem (model) bestaan, noemen we de structuur (ordening) van het systeem.

De term relatie wordt in meerdere betekenissen gebruikt, namelijk:

- in de betekenis van *keten* of *volgorde*, in de zin van 'gaat vooraf aan' of 'volgt op';
- als *functie*, bijvoorbeeld $y = f(x)$;
- in de betekenis van *oorzakelijk verband*, zoals: y is het gevolg van x (causale relatie);
- als *procesrelatie*, in de zin van afdeling X 'moet leveren aan' afdeling Y.

In sommige vakgebieden wordt het begrip relatie gebruikt als synoniem voor functie. Men spreekt dan wel van een functionele relatie. Naast dit soort relaties komen bij simulatie ook relaties in de andere betekenissen voor.

3. PARAMETERS

Behalve de vier soorten variabelen en de relaties hiertussen hebben we ten behoeve van een systeembeschrijving ook nog *parameters* nodig. Hiervan onderscheiden we twee groepen, namelijk:

1. de parameters waardoor functionele relaties worden beschreven;
2. de parameters waardoor de verdelingsfuncties van de stochastische variabelen worden beschreven.

Voorbeelden van de eerste groep vinden we in een relatie als:

$$Q = a \cdot L^{b} \cdot K^{1-b}$$

waarin: Q = produktie
L = hoeveelheid arbeid
K = hoeveelheid kapitaal
a en b = parameters, waarmee de betekenis van L en K 'op' Q kan worden uitgedrukt.

Tot de tweede groep van parameters worden die grootheden gerekend, waardoor de verdelingsfuncties F(x) van de stochastische variabele x 'volledig gekend en begrepen' worden. Een voorbeeld hiervan vinden we in de verdelingsfunctie:

$$F(x) = P(\underline{x} \leqq x) = 1 - e^{-\lambda x} \qquad (x > 0, \ \lambda > 0)$$

waarin λ de parameter voorstelt. Als λ bekend is kan voor elke waarde van x $(x > 0)$ een kans worden berekend.

Op het onderwerp parameterschatting komen we terug in hoofdstuk III.

Bij het ontwerpen van modellen hebben we dus te maken met variabelen, relaties en met parameters. Gezamenlijk moeten ze een beeld geven van de interne structuur van het te onderzoeken systeem en van de samenhang met de relevante omgeving van het systeem.

II.7 HET DOEL EN HET GEBRUIK VAN SIMULATIEMODELLEN*)

Tot slot van de bespreking van het modelbegrip gaan we nog kort in op de zin en het gebruik van simulatiemodellen. Uit het voorgaande zal duidelijk zijn geworden dat bij het proces van kennisverwerving omtrent de werkelijkheid, de noodzaak van afbeelden en abstractie evident is. Het proces van kennisverwerving is in feite het formuleren van verklaringen en het toetsen hiervan. Wij definiëren dat proces als het construeren van een model: het model vormt de samenvatting van de verworven kennis. Anderen gebruiken hiervoor het begrip 'theorie', hetgeen in deze betekenis dus identiek is met 'model'.

Het is evenwel van belang om te beseffen, dat daarnaast het *gebruik* van modellen kennis omtrent de werkelijkheid kan geven. We beperken ons hier bewust tot 'model', omdat het nu vooral gaat om de toepassingssfeer.

De functie van een model bij het verkrijgen van kennis wordt vooral ontleend aan de mogelijkheid tot *experimenteren met een model.*

Als zodanig geeft het model de mogelijkheid het gezegde 'bezint, eer ge begint' ten uitvoer te leggen.

Het experimenteren met het model is goedkoper en eenvoudiger uitvoerbaar dan het rechtstreeks ingrijpen in de werkelijkheid. Dat laatste is overigens zelfs vaak niet mogelijk.

Het is bijvoorbeeld niet verantwoord om de prijzen van levensmiddelen gedurende één maand te verdubbelen, om zodoende de uitwerking van deze prijsmaatregel op het koopgedrag van het publiek te onderzoeken.

Het is wel mogelijk om een model te construeren, waarmee dit experiment, eventueel met behulp van de computer, kan worden uitgevoerd. Zo verkrijgt men door simulatie-experimenten informatie, waarvan men hoopt dat deze zo goed mogelijk aansluit bij de werkelijkheid. Het zal duidelijk zijn dat de 'kwaliteit' van dergelijke informatie afhankelijk is van de 'kwaliteit' van het model.

Anders gezegd: de getrouwheid van de afbeelding is bij simulatie-experimenten van essentieel belang.

Overigens maakt het natuurlijk wel uit welke doelstelling bij het maken van en het experimenteren met een model wordt gehanteerd. Als bijvoorbeeld een model gebruikt moet worden voor voorspellingen, zal in het algemeen een grotere nauwkeurigheid gewenst zijn dan wanneer het gaat om een model waarmee een keuze moet worden gemaakt uit diverse alternatieve systemen.

---

*) Sommige opmerkingen zijn eveneens van toepassing op modellen in het algemeen.

Simulatie-experimenten worden uitgevoerd met een grote verscheidenheid aan doelstellingen. We noemen er een aantal.

1. Evaluatie: hoe goed voldoet een bepaald systeemontwerp, als het wordt beoordeeld op grond van een aantal door de opdrachtgevers te hanteren criteria?

2. Vergelijking: bij het ontwerpen van alternatieve systemen (oplossingsalternatieven), waarbij bepaalde functies moeten worden verricht, doet het model dienst om snel de alternatieven door te rekenen en te evalueren.

3. Voorspelling: met behulp van experimenteren worden de prestaties van een systeem, onder een bepaalde verzameling (mogelijk toekomstige) condities geschat.

4. Gevoeligheidsanalyse: bepaal welke van de vele factoren het belangrijkst zijn voor de werking van het systeem; hoe gevoelig is het systeem voor kleine variaties in die factoren; m.b.v. het model kunnen 'what-if'-vragen gesteld worden.

5. Optimalisatie: bepaal exact welke combinatie van factoren de beste totaaluitkomsten van het systeem opleveren. Simulatie (waarbij dit per definitie niet zonder meer kan) leidt tot het onderkennen van functionele relaties, waarmee een analytisch optimaliseringsmodel kan worden opgebouwd.

6. Structuuronderzoek: wanneer van een systeem de werking niet bekend is, dus voor ons een black-box is, dan kan simulatie de black-box openen; dat is de interne structuur opleveren.

7. Coördinatie: het oplossen van afstemmingsproblemen, bijvoorbeeld door het meenemen van extra beperkingen, of het *simultaan* bepalen van meerdere endogene variabelen.

8. Integratie: met behulp van modellen als onderdeel van mens-machine-systemen kunnen we een deel van de horizontale en verticale integratie realiseren. (horizontaal: sub-optimalisatie tegengaan; verticaal: afstemming over managementniveaus)

Deze lijst is niet uitputtend; wel kan men eruit afleiden, dat het zinvol is om de eigen doelstellingen goed te overdenken.

II.8 OPGAVEN

Opgave 1

Welk type relaties - beschreven in § II.6 - kent het model van de haringkar uit § I.3? Zou Simulatie voor de haringkar de meest geschikte methode van onderzoek zijn?

Opgave 2

Maak voor de haringkar een dynamisch model door rekening te houden met seizoensinvloeden en de dag in de week.

Opgave 3

Een bekend probleem is dat van 'De dronken zeeman'.
Deze is na het passagieren in de haven op zoek naar zijn schip.
In de moderne havenwijk zijn alle straten in vierkante blokken gesitueerd. Staande op de hoek van een straat moet hij beslissen of hij noord of zuid, dan wel oost of west gaat. Deze beslissing herhaalt zich bij elke straathoek. Wat is de kans dat hij na 10 blokken gewandeld te hebben zich op niet meer dan 2 blokken van zijn startpunt bevindt? Hierbij is aangenomen dat hij geen enkele voorkeur voor een bepaalde richting heeft bij zijn keuze op een hoek.

1. Ontwerp een model waarmee dit probleem m.b.v. simulatie kan worden opgelost.
2. Zet het model om in een computerprogramma in de taal van uw keuze en laat uw (micro)computer de gestelde vraag beantwoorden.

Opgave 4

In een eenvoudig model van een personeelssysteem komen o.a. de volgende grootheden aan de orde:
- aantal werknemers
- toename of afname van het personeelsbestand (uitbreiding of inkrimping)
- fractie van de gewenste personeelsverandering (vermeerdering of vermindering)
- arbeidscapaciteit
- reserve capaciteit (overuren)
- hoeveelheid te verrichten werk op korte en lange termijn (prognose)
- achterstallig werk
- personeelskosten
- waarde kapitaalgoederen
- functie specificaties
- arbeidsmarkt
- enzovoorts.

1. Beschrijf het model met behulp van een schets en classificeer de verschillende typen variabelen.
2. Kunt u ook parameters in het model onderkennen?
3. Geef aan hoe u de relaties specificeert.
4. Pas de lijst uit § II.7 toe op het personeelsmodel en bespreek het gebruik.

HOOFDSTUK III

Het Ontwikkelen van een Simulatiemodel

III.1 INLEIDING

Een simulatie-onderzoek is opgebouwd uit verschillende stappen. Deze stappen verlopen overeenkomstig de fasen van elk wetenschappelijk onderzoek: inductie, deductie en validatie.

In § 2 geven we de indeling van een simulatie-onderzoek die in het vervolg van dit hoofdstuk alsmede in de hoofdstukken IV t/m VI wordt uitgewerkt. De hoofdstukken III t/m VI vormen dus een onlosmakelijk geheel.

In dit hoofdstuk staat de modelbouw centraal. In § 3 bespreken we de eerste stappen van het onderzoek, namelijk het vaststellen van de doelstelling(en) en de probleemformulering. Deze stappen moeten leiden tot een beeldconstructie. Hiervan geven we een voorbeeld, namelijk een voorraadbeheersingsprobleem in § 4.

Een essentiële voorwaarde voor simulatie-onderzoek is de mogelijkheid trekkingen te kunnen doen uit verdelingsfuncties. De technische omstandigheden om deze te realiseren zijn door de ontwikkelingen op het gebied van de automatisering goed. De methoden die de trekkingen mogelijk maken worden aangeduid als Monte Carlo-procedures. Dit onderwerp wordt besproken in § 5. Bovendien gaan we in deze zelfde paragraaf in op methoden om te komen tot schattingen van parameters in functionele vergelijkingen.

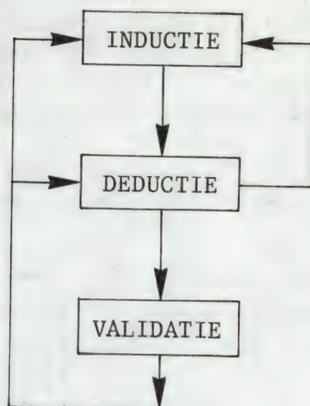
Als laatste stap van de modelontwikkeling moet het model worden geëvalueerd. Daar simulatiemodellen vaak ingewikkeld zijn, mede door de aanwezigheid van verdelingsfuncties kan de modelevaluatie in deze fase niet volledig zijn. Pas nadat een computerprogramma is gemaakt kan een volledige evaluatie worden uitgevoerd.

Gezien het feit dat de volgende fasen van het onderzoek veel tijd vragen en mede door het gebruik van de computer kostbaar zijn, willen we benadrukken dat zelfs een beperkte evaluatie zeer essentieel is. Aan de modelevaluatie besteden we aandacht in § 6.

Het hoofdstuk wordt besloten met een aantal opgaven in § 7.

III.2 HET SIMULATIE-ONDERZOEK IN FASEN: DE MODELCYCLUS

Modelbouw vindt plaats in fasen. Het proces om tot een model te komen wordt wel aangeduid als *modelcyclus*. In de literatuur worden verschillende beschrijvingen van de modelcyclus gegeven. De meest algemene is de beschrijving zoals die in figuur III.2a is gegeven.



Figuur III.2a

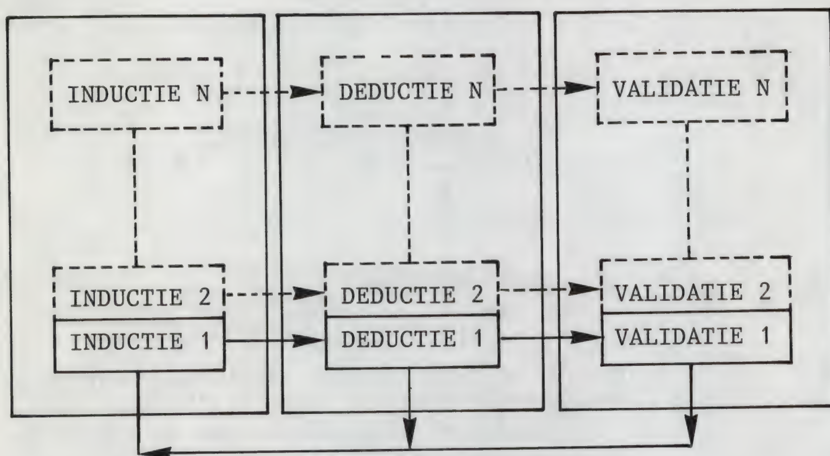
Met *inductie* wordt de eigenlijke modelbouw bedoeld. Hiertoe wordt het reële systeem waarvan men een model wil bouwen bestudeerd. De variabelen en de relaties worden beschreven om een zo goed mogelijke, op het probleem afgestemde weergave te krijgen van het reële systeem. Het proces van modelontwikkeling wordt in de volgende paragrafen van dit hoofdstuk uitgewerkt.

Een model is een middel om het gedrag van een reëel systeem te verklaren. In de *deductie*-fase worden experimenten met het model uitgevoerd om na te gaan of dit een voldoende representatieve afbeelding is van de realiteit. Dit kan bijvoorbeeld door met het model te rekenen en de resultaten hiervan te vergelijken met resultaten uit de werkelijkheid. Als deze vergelijking, ook na herhaalde experimenten positief uitvalt, kan het model worden aanvaard als een adequate beschrijving van het reële systeem en worden toegepast in de werkelijkheid (de realisatie). In de hoofdstukken IV en V wordt de deductiefase nader uitgewerkt.

Naast de inductie en de deductie maakt de *validatie* deel uit van de modelcyclus. Een valide model is een model op grond waarvan conclusies getrokken kunnen worden die in de werkelijkheid waarvoor het model een verklaring moet geven, gebruikt kunnen worden. Dit proces van validatie omvat veel meer dan in een figuur tot uitdrukking kan worden gebracht. Het proces voltrekt zich in

diverse stappen zowel tijdens als na de inductie- en deductiefase. Een meer uitgebreide beschrijving van validatie wordt gegeven in hoofdstuk VI.

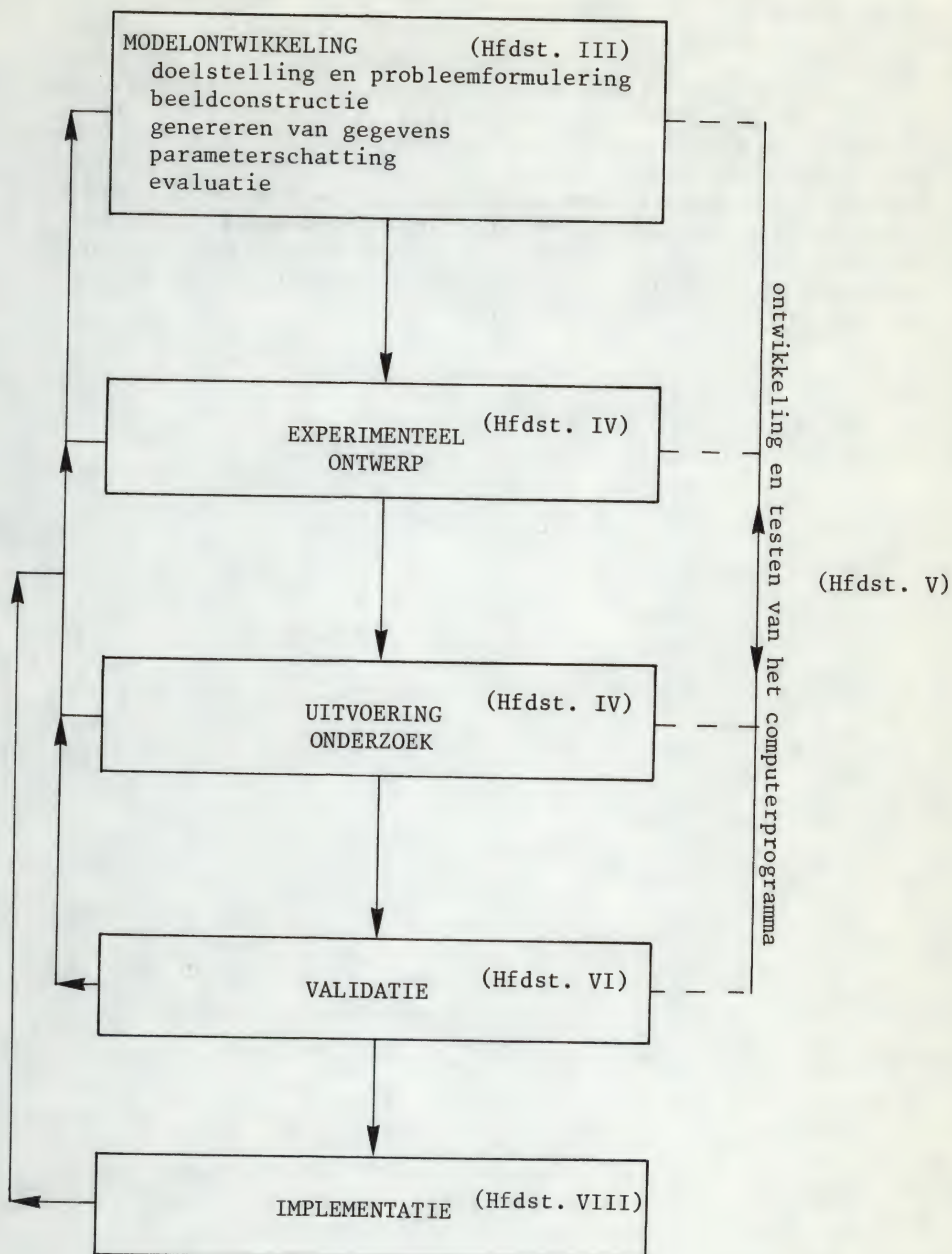
In de praktijk van een onderzoek verlopen trouwens alle drie de fasen vaak verbrokkeld. De onderzoeker heeft bepaalde ideeën met betrekking tot de werkelijkheid; hij heeft een beeld van het te onderzoeken concrete systeem. Op grond van dit beeldsysteem zal hij een model construeren, gegevens verzamelen en het model oplossen. Als de resultaten van de oplossing niet in overeenstemming zijn met de werkelijkheid zal het model worden aangepast en opnieuw worden 'doorgerekend'. Dit proces zal zich voortzetten totdat hij vindt dat het model het gedrag van het systeem voldoende getrouw weerspiegelt. Figuur III.2b is een poging dit proces realistischer weer te geven.



Figuur III.2b

Hoewel het hier geschetste proces van 'vallen en opstaan' nooit geheel vermeden kan worden, zal een onderschatting van de eerste fase tot nadelige gevolgen in de volgende fasen leiden. Een effectief onderzoek staat of valt met een correcte probleemformulering en modelconstructie. We komen hierop in de volgende paragrafen terug.

De nadere detaillering van de drie fasen is zoals gezegd in de literatuur nogal verschillend. In dit boek zal de indeling worden aangehouden zoals in figuur III.2c is weergegeven. Het eerste blok van deze figuur komt overeen met de inductiefase, de blokken twee en drie (experimenteel ontwerp en uitvoering onderzoek) vormen de deductiefase en het blok validatie komt overeen met het gelijknamige blok uit figuur III.2a. De implementatie tenslotte is toegevoegd, omdat een simulatieproject pas geslaagd kan worden genoemd, als het model ook daadwerkelijk in de praktijk wordt ingevoerd en gebruikt. Met het oog op een succesvolle implementatie bespreken we in hoofdstuk VIII enkele aspecten van projectbesturing.



Figuur III.2c Fasering simulatie-onderzoek

We hebben in figuur III.2c het ontwikkelen en testen van het computerprogramma gezet naast de eigenlijke fasering van het onderzoek. Hierdoor willen we aangeven dat de computer in het algemeen bij elk van de fasen als hulpmiddel kan worden ingezet. De ontwikkeling van simulatietalen is van dien aard dat het computerprogramma al direct kan worden mee-ontwikkeld. Dit betekent dat de volgorde van onderzoek die in de figuur gesuggereerd wordt minder nadrukkelijk is. Zo zijn bepaalde zaken die bij het experimenteel ontwerp aan de orde komen reeds relevant bij het blok 'modelontwikkeling'. Met name geldt dit voor de evaluatie van het model.

Op grond van didactische overwegingen behandelen we de verschillende fasen gescheiden. In dit hoofdstuk zal daarom aan het blok modelontwikkeling aandacht worden besteed, zonder dat over het computerprogramma wordt gesproken.

III.3 VASTSTELLING DOELSTELLINGEN EN PROBLEEMFORMULERING

Probleemformulering is een essentieel onderdeel van een onderzoek. Het verkrijgen van relevante, zo volledig mogelijk en betrouwbare informatie over het te onderzoeken probleem is een cruciale voorwaarde voor succes. Uit de praktijk en de literatuur komt vaak de klacht, dat oplossingen niet op het probleem zijn afgestemd. Vaak wordt veel te snel door onderzoekers aangenomen, dat het probleem duidelijk is. Shannon merkt in dit verband op, dat er jaarlijks miljoenen dollars worden verspild door schitterende en elegante oplossingen te creëren voor de verkeerde problemen.

Nu is het zeker niet zo, dat de oorzaak van de verspilling alleen bij de onderzoeker ligt. Ook de probleemhebbers blijven als informatieverschaffers vaak in gebreke. Davis*) noemt als belangrijke reden hiervoor de beperkingen van de mens in zijn mogelijkheden tot het verwerken en selecteren van informatie en tot het oplossen van problemen. Het menselijke geheugen is te onderscheiden in twee delen, een korte-termijn en een lange-termijn geheugen. Het eerste deel is klein en kent een korte opslagduur. Bij een kortstondige en beperkte probleemanalyse zullen deze nadelen een negatieve invloed hebben op een adequate probleemformulering. De verstrekte informatie zal sterk afhangen van het moment waarop deze versterkt wordt. Het opdiepen uit het langetermijn geheugen vraagt tijd en wordt beïnvloed door de omstandigheden waaronder dit wordt geraadpleegd.

Behalve de beperking van het menselijke geheugen zijn er andere oorzaken die een goede probleemformulering in de weg staan. In hoofdstuk VIII komen we hierop terug. Ook zullen we in dat hoofdstuk ingaan op hulpmiddelen om tot een succesvolle probleemanalyse en -formulering te komen. In deze paragraaf zullen we ons bezig houden met de inhoudelijke kant van de probleemformulering.

In het algemeen kan worden gesteld, dat bij elke probleemanalyse in eerste instantie twee vragen beantwoord moeten worden:

a. wat is de doelstelling? en
b. welke zijn de hindernissen die het bereiken van de doelstellingen in de weg staan?

Teneinde op deze vragen de juiste antwoorden te kunnen vinden is een goed begrip van het systeem, waarop het probleem betrekking heeft een eerste vereiste.

Wanneer we onze (vaak gebrekkige) inzichten omtrent het systeem enige structuur wensen te geven, mede ter voorbereiding op het maken van een model, dan moeten de volgende stappen worden uitgevoerd:

---

*) G.B. Davis, Strategies for information requirements determination, IBM Systems Journal 21, 1(1982), pg. 4-30.

1. De probleemspecificatie en de specificatie van de doelstellingen van het model.
2. Het formuleren van het kwalitatief model.
3. Het specificeren van een voorlopige verzameling variabelen en parameters.
4. Het specificeren van de relaties tussen de variabelen in de vorm van hypothesen.
5. Met behulp van de waarnemingsgegevens formuleren van een kwantitatief model, waarbij de relaties en parameters van inhoud worden voorzien.

Let wel: In deze fasering is voortdurende terugkoppeling noodzakelijk; modelbouw is een iteratief proces.

Vooral het vaststellen van het doel of van de doelstellingen is een niet te onderschatten onderdeel van de modelbouw. Het doel heeft namelijk gevolgen voor elk van de volgende stappen van het onderzoek. Niemand zou een model moeten bouwen zonder eerst een goed gedefinieerd doel te formuleren. Het hele proces van modelontwerp, validatie, experimenteel ontwerp en het trekken van conclusies is nauw gerelateerd aan het gestelde doel.

Zo worden er, als het doel van de modelbouw is een nieuw of bestaand systeem te evalueren, hoge eisen gesteld aan de accuraatheid van het model; er is dan een hoge graad van gelijkvormigheid met het reële systeem vereist.

Er bestaan nogal wat uiteenlopende doelstellingen t.a.v. het gebruik van modellen, zoals we in § II.7 tot uitdrukking hebben gebracht. Zoals ze daar geformuleerd staan, is de omschrijving voor specifieke problemen te globaal. De doelstellingen zullen dan veel concreter aangegeven moeten worden. Het ligt voor de hand dit in overleg met de opdrachtgevers te doen. Een doelstellingenanalyse zal voor alle betrokken partijen zeer verhelderend kunnen werken. De onderzoeker zal zodoende tot een beter begrip van de knelpunten kunnen komen. En om de oplossing van de knelpunten gaat het uiteindelijk.

Na vaststelling van het doel of de doeleinden moeten de relevante variabelen worden opgespoord en benoemd. Dat dit geen eenvoudige zaak is, zal blijken bij lezing van het volgende voorbeeld.

VOORBEELD

Een producent van elektronische artikelen is ruim voorzien van orders. Het grootste percentage van de orders gaat evenwel niet op tijd de deur uit. De consequenties van deze te lange levertijden laten zich gemakkelijk raden: op de korte termijn veel telefoontjes van boze afnemers, of de langere termijn wellicht een verlies van orders. Het probleemgebied is de orderafhandeling; het knelpunt

is de verzending die in veel gevallen te laat gebeurt.

De oorzaken van deze problematiek kunnen vele zijn. We noemen hieronder een aantal in een willekeurige volgorde:

- Foutieve prognoses van de afzet;
- Te kleine produktiecapaciteit;
- Slechte informatiesystemen, tot uitdrukking komend in bijvoorbeeld een gebrekkige communicatie tussen de afdelingen verkoop en produktie of in het ontbreken van een levertijdbewakingssysteem;
- Grote afhankelijkheid van toeleveranciers;
- Foutieve produktieplanning;
- Verouderende machines, hetgeen leidt tot grotere uitvalpercentages;
- Gebrek aan vakbekwaam personeel;
- Slechte motivatie bij het personeel;
- Hoog ziekteverzuim op de produktie-afdeling.

Stel dat als doelstelling van een onderzoek voor de hier geschetste situatie de volgende formulering gekozen wordt:

> "Ontwerp een orderafhandelingssysteem, zodanig dat gemiddeld x% (bijvoorbeeld x = 95) van de orders binnen de met de leveranciers overeengekomen tijd afgeleverd wordt."

De vraag die de onderzoeker zich nu moet stellen is welke factoren met het oog op realisatie van deze doelstelling van een zodanig belang zijn, dat ze bij het onderzoek betrokken dienen te worden. Gezien de vele factoren die hierboven genoemd zijn als mogelijke oorzaken van de problematiek is een grondige analyse wenselijk. Deze analyse dient erop gericht te zijn de meest relevante factoren op te sporen.

In deze fase van het onderzoek worden we dus geconfronteerd met de vraag: welke variabelen betrekken we in het onderzoek en welke laten we daar buiten. Terwijl aan de ene kant gewaarschuwd moet worden voor oppervlakkigheid van probleembenadering en de drang om tot een al te snelle oplossing van de te onderzoeken problematiek te komen, moet aan de andere kant ook worden opgepast voor te veel variabelen en relaties. Naarmate meer variabelen nodig zijn om een probleem te beschrijven en te verklaren, zal het moei-lijker zijn het probleem te doorzien. Ook de resulaten van de
oplossing zullen moeilijker te interpreteren zijn. Hier speelt dus een lastig afwegingsprobleem voor de onderzoeker, waarvoor geen ééndidige oplossingen te geven zijn.

Uit de door ons gehanteerde definitie van simulatie blijkt, dat om oplossingen van problemen aan te kunnen dragen verklaring van het gedrag van het reëele systeem noodzakelijk is. Dit betekent, dat de structuur van het reëele systeem bekend zal moeten worden. Simulatie kan ons hierbij helpen. Simulatie kan inzicht geven in het

transformatieproces dat waarden van de invoervariabelen omzet in waarden van de uitvoervariabelen. Een modelspecificatie zal daarom moeten omvatten:

- een beschrijving van de interne structuur (de elementen en de relaties tussen de elementen);
- een beschrijving van de externe structuur (de relevant geachte exogene variabelen en de invoerspecificaties).

Ter toelichting op het voorgaande geven we in de volgende paragraaf een voorbeeld.

III.4 BEELDCONSTRUCTIE EN MODELFORMULERING; EEN VOORBEELD

Voorbeelden van praktijkproblemen die dikwijls m.b.v. simulatie worden geanalyseerd liggen op het terrein van de wachttijd- en de voorraadsystemen. De wiskundige modellen van deze systemen worden al vrij snel zo ingewikkeld, dat analytische oplossingen moeilijk te realiseren zijn. In deze paragraaf zal ter illustratie van de behandelde theorie een voorraadmodel nader worden uitgewerkt. Het gaat om een voorraadbeheersingsprobleem van een groothandel, waarbij als doelstelling wordt gehanteerd:

- het streven naar zo laag mogelijke kosten, waarbij met kosten worden bedoeld de som van voorraadkosten, kosten van 'neen-verkopen' en bestelkosten.

In de terminologie van § II.6 zijn de totale kosten (TK) aan te duiden als een endogene variabele, en wel als een uitgangsvariabele.

Behalve in de totale kosten zijn we ook geïnteresseerd in de servicegraad (SG) van het systeem. Met de servicegraad bedoelen we het percentage van de bestellingen dat direct uit voorraad geleverd kan worden. Het nastreven van een bepaalde servicegraad zou als een nevendoelstelling aangemerkt kunnen worden. We beschouwen de variabele servicegraad dus ook als een uitgangsvariabele, maar één die ondergeschikt is aan de variabele totale kosten.

De totale kosten alsmede de servicegraad staan onder invloed van een aantal exogene variabelen, enerzijds omgevingsvariabelen en anderzijds beslissingsvariabelen of instrumentele variabelen.

Allereerst zal het voorraadniveau en daarmee ook de kostenfactor beïnvloed worden door de vraag naar het produkt.

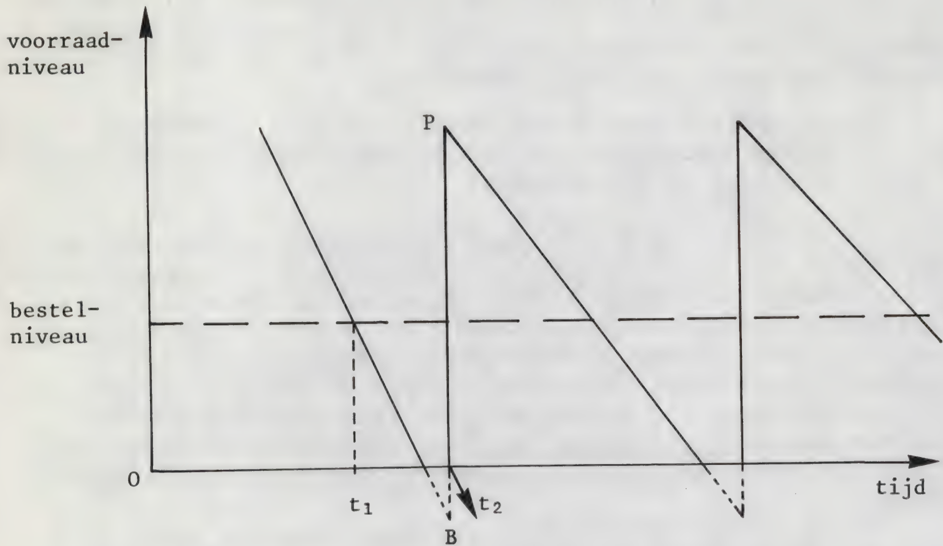
De vraag D naar het produkt Z beschouwen we als een stochastische variabele met een bepaalde waarschijnlijkheidsverdeling f(D). De bepaling van de vorm van de verdeling met de bijbehorende parameters behoort tot de volgende deelfase van het onderzoek (zie § III.5).

Ook de levertijd (LT) van het produkt beïnvloedt de voorraad en daarmee de kosten hiervan. Als het produkt wordt besteld bij de leverancier gaat er enige tijd voorbij alvorens de bestelling binnenkomt. Deze levertijd blijkt verder bij nader onderzoek niet vast te zijn, maar variabel binnen bepaalde grenzen. Het lijkt daarom aannemelijk ook de levertijd als een stochastische variabele te beschouwen.

Terzijde willen we nog opmerken dat de groothandel de beide genoemde variabelen nooit als helemaal onbeïnvloedbaar zal beschouwen. Zowel de vraag naar een produkt als de levertijd kunnen, zij het in een geheel andere mate en op een geheel andere wijze als de instrumentele variabelen worden beïnvloed.

Met de 'echte' instrumentele variabelen kan door de organisatie

direct worden gemanipuleerd. In ons voorbeeld denken we bij dit soort van variabelen aan de bestelhoeveelheid (BQ) en het bestelniveau (BN). De bestelhoeveelheid heeft, zoals duidelijk zal zijn betrekking op het aantal eenheden dat in één keer bij de leverancier wordt besteld, terwijl het bestelniveau duidt op het voorraadniveau van het produkt dat tot de beslissing 'bestellen' aanleiding geeft. In het volgende plaatje van een voorraadmodel komen de diverse grootheden duidelijker tot uitdrukking.



Figuur III.4a Voorraadmodel

Op het moment t_1 is de voorraad gezakt tot het bestelniveau. Op dat moment wordt er een order geplaatst van de omvang BP. Als gevolg van de levertijd wordt deze order pas afgeleverd op het moment t_2.

Met behulp van een relatiediagram kunnen we de verschillende relaties weergeven zoals die in het geschetste model van belang zijn.

	TK	SG	D	LT	BQ	BN
TK	-	-	-	-	-	-
SG	x	-	-	-	-	-
D	x	x	-	-	x	x
LT	x	x	-	-	-	x
BQ	x	x	-	-	-	-
BN	x	x	-	-	-	-

Figuur III.4b Relatiediagram voorraadmodel

In een relatiediagram noteren we zowel langs de rijen als langs de kolommen alle variabelen die in het model aanwezig zijn. Vervolgens gaan we voor elke combinatie van twee variabelen na of er een relatie zou kunnen bestaan en zo ja, in welke richting.

Zo'n relatiediagram heeft als bijkomend voordeel dat hiermee gemakkelijker het overleg kan worden uitgevoerd met de opdrachtgevers. Het is namelijk eenvoudig leesbaar en zonder dat de opdrachtgever zich in allerlei, wellicht minder eenvoudige wiskundige formules hoeft te verdiepen kan hij meepraten over de relaties.

Het voorraadbeheersingsprobleem zou als volgt in een voorlopig mathematisch model kunnen worden weergegeven:

$$TK = f(BQ, BN, LT, D),$$

waarin TK moet worden geminimaliseerd*) om tot een optimale BQ en BN te komen.

Bij de beeldconstructie hebben we ook te maken met parameters. Allereerst hebben we parameters nodig voor de kostenvergelijking. De kosten van het voorraadsysteem kunnen worden onderscheiden naar:

- kosten van 'in voorraad hebben' $= c_1$ per eenheid per dag
- kosten van 'neen-verkopen' $= c_2$ per eenheid
- bestelkosten $= c_3$ per order

De totale kosten worden berekend op grond van de vergelijking

$$TK = TK_1 + TK_2 + TK_3$$

Deze vergelijking is tevens één van de relaties in het model.

Recapitulerend en aanvullend kunnen we dan de volgende beschrijving geven van het voorraadbeheersingssysteem zoals ons dat voor ogen staat.

1. *Omgevingsvariabelen*

D = stochastische variabele voor de vraag
LT = stochastische variabele voor de levertijd

*) De termen 'minimalisatie' en 'optimalisatie' lijken in tegenspraak met het gestelde in § 3. Ze zijn hier gerechtvaardigd omdat dit voorraadmodel geïsoleerd van andere bedrijfsbeslissingen is opgesteld (het zogenaamde klassieke model). In de praktijk zal men ook hier met aspiratieniveaus werken als men coördinatiefactoren meeneemt, zoals het seriegrootteprobleem bij de produktie, omstelkosten, klantencategorieën (m.b.t. de servicegraad), het liquiditeitsprobleem, enzovoorts. Daarom zal in een realistisch voorraadmodel het kostencriterium niet als belangrijkste doelvariabele in aanmerking mogen komen.

2. *Beslissingsvariabelen*

BQ = bestelhoeveelheid
BN = bestelniveau

3. *Uitgangsvariabelen*

TK = totale kosten
SG = servicegraad

4. *Toestandsvariabelen*

V = voorraadniveau
TK_1 = totale voorraadkosten
TK_2 = totale kosten van 'neen-verkopen'
TK_3 = totale bestelkosten

5. *Parameters*

- c_1, c_2 en c_3
- parameters van de verdelingsfuncties
  bijvoorbeeld μ en σ bij de normale verdeling
  n en p bij de binomiale verdeling

6. *Relaties*

$$(TK_1)_t = \tfrac{1}{2}c_1 \cdot (V_t + V_{t-1}) \cdot n \qquad V_{t-1} \geqq V_t$$

$$V_t \geqq 0$$

n = aantal dagen in periode t
of zoveel minder als de voorraad in de loop van
periode t negatief wordt

$$(TK_2)_t = c_2 \cdot x_t$$

x_t = aantal eenheden in periode t dat wegens onvoldoende voorraad niet geleverd kan worden en afbesteld wordt

$$(TK_3)_t = c_3 \cdot y_t$$

y_t = aantal bestellingen in periode t

$$TK = \sum_t \{(TK_1)_t + (TK_2)_t + (TK_3)_t\}$$

$$SG = 1 - \frac{\sum_t x_t}{TV}$$

TV = totale vraag over alle perioden t

In deze relaties komen variabelen voor die nog niet genoemd zijn, namelijk n, x_t, y_t en TV. Hiervan zijn n, y_t en TV hulpvariabelen die in een computerprogramma op eenvoudige wijze bijgehouden kunnen worden. Zij kunnen tot de toestandsvariabelen worden gerekend.

De variabele x_t kan als een stochastische omgevingsvariabele in het model worden ingebracht maar ook als een vast percentage van de vraag waaraan wegens onvoldoende voorraad niet onmiddellijk kan worden voldaan en dat wordt afbesteld.

Met dit voorbeeld hebben we willen demonstreren hoe een simulatiemodel tot stand kan komen. Alvorens met het model te kunnen experimenteren, moeten o.a. nog gegevens over de omgevingsvariabelen en de parameters worden verzameld. Met behulp van de in de volgende paragraaf te bespreken technieken kunnen deze gegevens worden vastgesteld.



----- het beeldsysteem -----

III.5 HET GENEREREN VAN GEGEVENS

De voor het model benodigde gegevens over de onafhankelijke variabelen kunnen op verschillende wijzen worden verkregen. In de statistiek worden bijvoorbeeld de enquête en de historie genoemd als gegevensbronnen. Op deze technieken gaan wij hier niet in. In deze paragraaf zullen we met name ingaan op de zogenaamde Monte Carlo-procedures. Met Monte Carlo-procedures worden methoden bedoeld die het mogelijk maken aselecte trekkingen te doen uit verdelingsfuncties. Door deze procedures kunnen in beginsel onbeperkt gegevens van verschijnselen met bekende waarschijnlijkheidsverdelingen worden gegenereerd. Juist deze procedures, die door de technische vooruitgang steeds beter toegepast kunnen worden, zijn voor simulatie-onderzoek een essentiële voorwaarde.

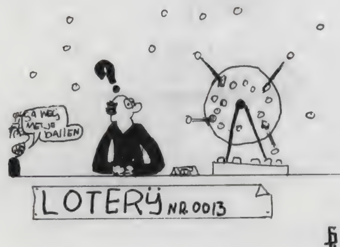
We zullen in deze paragraaf achtereenvolgens iets zeggen over toevalscijfers, over trekkingen uit verdelingsfuncties alsmede over het verband tussen de empirische gegevens en de Monte Carlo-procedures.

III.5.1 Toevalsgetallen

Toevalsgetallen of aselecte getallen ('random numbers') zijn het resultaat van een stel onderling onafhankelijke trekkingen uit een uniforme of rechthoekige verdeling over het interval (0,1). Omdat aan het afleiden van deze getallen door een computerprogramma altijd een deterministische procedure ten grondslag ligt, spreekt men wel van 'pseudo random numbers'.

De verschillende methoden voor het produceren van toevalsgetallen zullen hier niet aan de orde worden gesteld. Voor een goed begrip van wat bedoeld wordt, geven we enkele voorbeelden.

Voor met de hand uitgevoerde simulaties kan voor het verkrijgen van een reeks toevalsgetallen 0, 1, 2, 3, ..., 9 bijvoorbeeld



- een krachtige random generator -

gebruik worden gemaakt van een 'rad van avontuur' dat verdeeld is in tien sectoren. Elk van de sectoren krijgt één van de nummers 0, 1, 2, 3, ..., 9 als opschrift. Als nu voor een experiment een reeks van 100 toevalsgetallen nodig is, kan het rad 100 keer worden gedraaid om de vereiste steekproefomvang te verkrijgen.

Andere mogelijkheden zijn een 20-vlaksdobbelsteen met op de zijvlakken de nummers 0, 1, 2, 3, ..., 9 elk tweemaal of een vaas met 10 gelijke knikkers, elk genummerd met één van de getallen 0, 1, 2, 3, ..., 9.

Daar de beschreven hulpmiddelen bij toepassing erg tijdrovend zijn, is gezocht naar formules die met gebruik van computerprogramma's snel een reeks van toevalsgetallen genereren.

Een veel toegepaste techniek bestaat uit het gebruik van 'lineaire congruentie generatoren'. De algemene gedaante van de formule voor deze generatoren kan in de vorm van een recursieve vergelijking worden geschreven als:

$$x_i = (ax_{i-1}+c) \bmod m$$

De waarden van a, c en m bepalen in belangrijke mate de kwaliteit van de randomgenerator. Voor m gebruikt men bij voorkeur een macht van 2 en een zo groot mogelijk getal. Dit laatste betekent, dat de grootte van de geheugenwoorden in de computer in feite de bovengrens bepaalt. Voor a heeft men gevonden dat een achtvoud ± 3 een 'goede' waarde is.

Een andere bekende, doch bepaald geen goede methode maakt gebruik van de 'rij van Fibonacci'. Hierbij ontstaat een getal door de voorgaande twee getallen uit een reeks te sommeren:

$$x_{n+1} = (x_n + x_{n-1}) \bmod m$$

In deze formule moeten x_0, x_1 en m worden gekozen. Voor $x_n = 0$, $x_1 = 1$ en $m = 100$ krijgen we de reeks:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 44, 33, 77, 10, 87, 97, - - - - - - - - - - - - - - , 97, 2, 99, 1, 0, 1, 1, 2, - - - - - - -

In dit voorbeeld herhaalt de reeks zich na 300 termen. Deze beperkte cycluslengte is niet alleen een gevolg van de keuze van de startwaarden voor x_0 en x_1 en de waarde van m, maar is in het algemeen een nadelige eigenschap van de reeks van Fibonacci. Een ander nadeel is, dat in de reeks na een even getal steeds twee oneven getallen volgen. Door deze nadelen voldoet deze procedure niet aan de eisen van een ideale random generator.

In de literatuur wordt een aantal eisen genoemd waaraan een getallenreeks moet voldoen om als pseudo toevalsgetallen te kunnen worden aangeduid. Een ideale generator van pseudo toevalsgetallen moet een reeks opleveren, die:

1. uniform verdeeld is;
2. statistisch onafhankelijk;
3. reproduceerbaar;
4. een aanvaardbare cycluslengte heeft;
5. snel geproduceerd kan worden;
6. een minimum aan geheugencapaciteit van een computer vereist.

Om na te kunnen gaan of een rij aselecte getallen inderdaad aan bepaalde criteria voldoet, is reeds in de jaren '30 een aantal toetsen ontwikkeld. Aan deze toetsen zijn in de loop van de tijd andere toegevoegd. Voor de gebruikers van random generatoren is kennis van het voorgaande van belang om bedacht te zijn op eventuele nadelige eigenschappen. Alvorens tot simulatie met ingewikkelde modellen over te gaan, moet men zich eerst van de kwaliteit van de generator overtuigen, niet in het minst omdat een random generator gewoonlijk 'computer-afhankelijk' is.

III.5.2 Aselecte trekkingen uit verdelingsfuncties

Aselecte getallen zijn van essentieel belang voor trekkingen uit andere verdelingsfuncties. We zullen aan de hand van voorbeelden achtereenvolgens ingaan op trekkingen uit discrete verdelingen en uit kansdichtheden.

Bij discrete verdelingen heeft men te maken met trapfuncties. De waarde van de functie is steeds over een interval constant en verandert alleen op gegeven vaste punten.

VOORBEELD

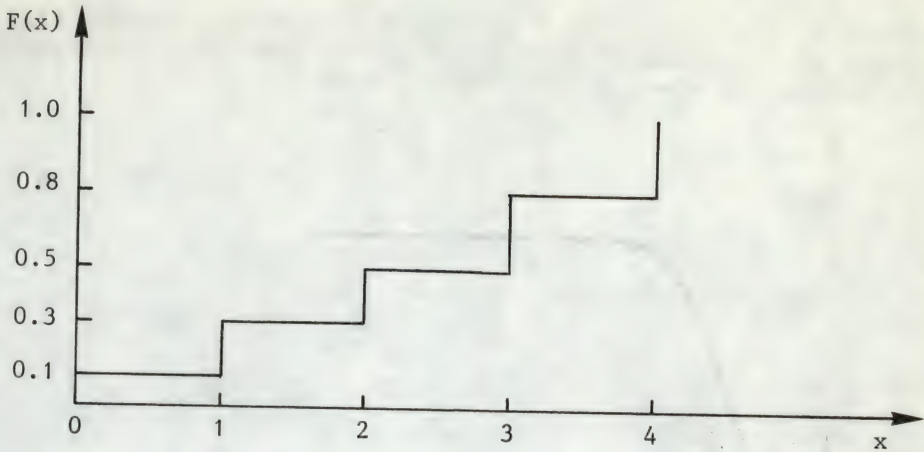
Gegeven is de kansverdeling:

x	0	1	2	3	4
f(x)	0.1	0.2	0.2	0.3	0.2

Hieruit is de volgende verdelingsfunctie af te leiden:

x	0	1	2	3	4
F(x)	0.1	0.3	0.5	0.8	1.0

In figuur III.5a is een grafische voorstelling van deze verdelingsfunctie getekend.



Figuur III.5a

Langs mechanische weg is een steekproef uit deze verdeling te realiseren door een reeks van aselecte getallen te trekken en vervolgens bij elk aselect getal de kleinste waarde van x te nemen waarvoor geldt:

$$r < F(x) = \sum_{x} P(\underline{x} = x_n)$$

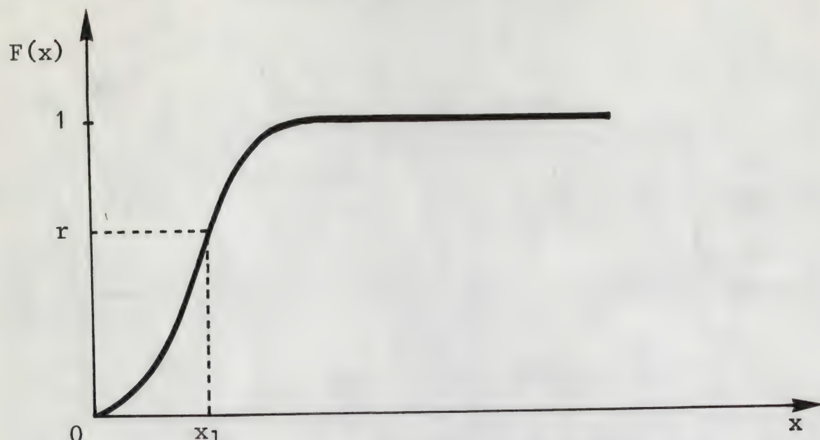
Als	$0 \leqq r < 0.1 \rightarrow x = 0$
als	$0.1 \leqq r < 0.3 \rightarrow x = 1$
als	$0.3 \leqq r < 0.5 \rightarrow x = 2$
als	$0.5 \leqq r < 0.8 \rightarrow x = 3$
als	$0.8 \leqq r < 1 \rightarrow x = 4$

In het algemeen kunnen uit een discrete verdeling aselecte trekkingen worden gedaan door eerst de verdelingsfunctie af te leiden, waarvoor geldt $0 \leqq F(x) < 1$ en vervolgens aan x, na de trekking van een aselect getal r de kleinste waarde toe te wijzen, die voldoet aan $r < F(x)$.

Voor trekkingen uit een kansdichtheid wordt principieel dezelfde procedure gevolgd. De moeilijkheid hierbij is soms de verdelingsfunctie. De algemene wiskundige formulering voor een verdelingsfunctie luidt:

$$F(x) = P(\underline{x} \leqq x) = \int_{-\infty}^{x} f(t)dt$$

waarin $0 \leq F(x) \leq 1$ en $f(t)$ de waarde van de dichtheidsfunctie is voor $x = t$. In figuur III.5b is een grafiek van een verdelingsfunctie voor een continue variabele getekend.



Figuur III.5b

Omdat $F(x)$ gedefinieerd is op het gebied $(0,1)$ kan een trekking uit de kansdichtheid worden gerealiseerd door na een trekking van een aselect getal r de bijbehorende x_1 te berekenen (zie voor een voorbeeld § I.3.4).

Voor het uitvoeren van simulatie-experimenten op de computer behoeven bij gebruik van simulatietalen niet meer, zoals in het verleden, speciale programma's voor het genereren van kansvariabelen te worden geschreven. Deze talen beschikken namelijk over speciaal hiervoor ontwikkelde procedures.

III.5.3 Empirische gegevens en Monte Carlo-procedures

Voor toepassing van de in de vorige deelparagrafen beschreven methoden is noodzakelijk, dat de verdelingsfunctie van de stochastische variabelen bekend is. Om hiervan op de hoogte te komen kan gebruik worden gemaakt van empirische gegevens, als deze tenminste bekend zijn voor de betrokken variabele.

Is dit laatste niet het geval, dan kan men twee dingen doen:

a. alsnog metingen voor de betrokken variabelen uitvoeren, of
b. een verdelingsfunctie veronderstellen.

De tweede mogelijkheid zal pas als laatste redmiddel mogen worden gebruikt. Zoals gezegd kan, om na te gaan welke verdelingen voor de onafhankelijke variabelen gelden, gebruik worden gemaakt

van empirische gegevens. Het vergelijken van een empirische en een theoretische frequentieverdeling, met als doel na te gaan of de theoretische verdeling een goede beschrijving geeft van de realiteit is een statistisch probleem.

Dit probleem kan bij eerste benadering worden bekeken met behulp van grafische voorstellingen in combinatie met berekening van een aantal parameters. De parameters van de theoretische verdeling worden vergeleken met de corresponderende parameters van de empirische verdeling.

Hieronder geven we de parameters van een aantal theoretische verdelingen.

Naam theoretische verdeling	Kansverdeling/ kansdichtheid	Parameters
binomiale	$f(x) = \binom{n}{x} p^x q^{n-x}$	n en p
Poisson	$f(x) = \frac{\lambda^x}{x!} e^{-\lambda}$	λ
negatief-exponentiële	$f(x) = \lambda e^{-\lambda x}$	$\frac{1}{\lambda}$
normale	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ en σ

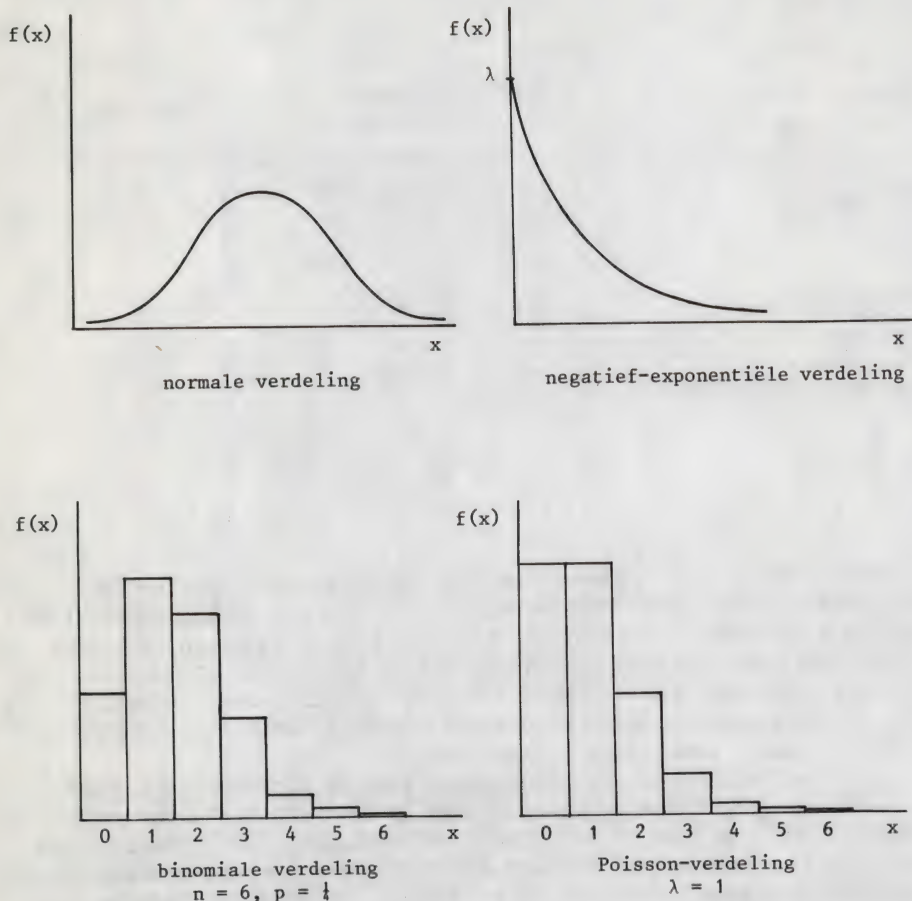
We berekenen dus de parameters van de gevonden empirische verdeling. Als na deze berekening blijkt, dat het gemiddelde in de omgeving ligt van de variantie, dan kan het zijn dat we te maken hebben met een Poisson-verdeling, als er tenminste bovendien sprake is van een discrete variabele. We waarschuwen er echter voor dat de bedoelde gelijkheid geen bewijs is voor een Poisson-verdeling, maar slechts een indicatie.

Ter aanvulling op de berekening van de parameters worden de gevonden gegevens uitgezet in een grafiek om zodoende een indruk te krijgen van de vorm van de verdeling. De verschillende theoretische verdelingen hebben elk hun eigen karakteristieke vorm. Door deze te vergelijken met de grafische voorstelling van de empirische gegevens kan misschien een volgende aanwijzing ontstaan omtrent de congruentie van de empirische verdeling met een bepaalde theoretische verdeling.

In figuur III.5c hebben we van een aantal theoretische verdelingen hun grafische voorstelling geschetst.

Als de aanwijzingen sterk in de richting van een bepaalde verdeling wijzen, kan tenslotte de hypothese worden getoetst dat de empirische gegevens inderdaad een steekproef vormen die

afkomstig is uit een bepaalde theoretische verdeling. Ten behoeve van deze toetsing kan gebruik worden gemaakt van de chi-kwadraat (χ^2) toets of van de Kolmogorov-Smirnov toets. Men spreekt in dit verband ook wel van 'goodness of fit' toetsen. Zeer algemeen kan worden gesteld dat de chi-kwadraat toets van toepassing is voor grote steekproeven ($n \geqq 100$), terwijl de Kolmogorov-Smirnov toets juist bij kleinere steekproeven ($10 \leqq n < 100$) gebruikt kan worden. Voor een bespreking van deze toetsen verwijzen we naar de appendix.



Figuur III.5c
Grafische voorstelling van vier theoretische verdelingen

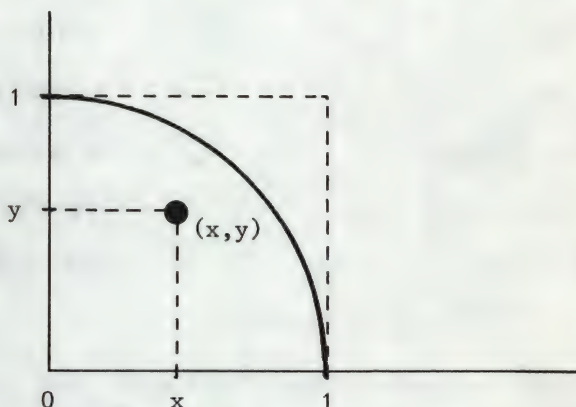
In deze paragraaf hebben we het gehad over toevalsgetallen, over trekkingen uit verdelingsfuncties en over de aanpassing van empirische verdelingen aan theoretische verdelingen. Deze onder-

werpen hangen nauw met elkaar samen. Als een (exogene) variabele door een kansverdeling beschreven kan worden, kunnen door gebruikmaking van de verdelingsfunctie en van toevalsgetallen willekeurige uitkomsten van de variabele worden gegenereerd. Dit genereren kan bovendien in onbeperkte mate gebeuren. Het nut van de theoretische verdelingen is van belang omdat, als een empirische verdeling in belangrijke mate overeenkomt met zo'n theoretische verdeling, dan voor het genereren van gegevens van zo'n theoretische verdeling gebruik kan worden gemaakt. Hiervoor kunnen dan namelijk standaardprocedures worden gebruikt.

De procedures die het mogelijk maken om willekeurig trekkingen uit verdelingsfuncties te doen, staan bekend onder de naam *Monte Carlo-procedures*. Voor de door ons in dit boek behandelde toepassingen van simulatie zijn deze procedures essentieel. Zij zijn zelfs een noodzakelijke voorwaarde voor het gebruik van de methode. Vandaar dat deze procedures in programmeertalen die speciaal ten behoeve van simulatie zijn ontwikkeld (zie hoofdstuk V) voor een aantal verdelingen standaard beschikbaar zijn. Ook andere programmeertalen, zoals Pascal en BASIC bieden wel mogelijkheden doordat zij in ieder geval over een procedure beschikken om toevalsgetallen uit het (0,1)-interval te trekken. De gebruiker kan dan zelf procedures schrijven voor andere verdelingsfuncties.

Ter illustratie en als afsluiting van deze paragraaf geven we een voorbeeld waarbij een deterministisch probleem met behulp van toevalsgetallen wordt opgelost.

In figuur III.5d ligt een kwart cirkel binnen het vierkant met lengte 1.



Figuur III.5d

Wanneer willekeurige paren toevalsgetallen (x,y) in het gebied (0,1) worden gegenereerd, zal een deel hiervan binnen de cirkel liggen. We noemen een paar (x,y) 'OK' wanneer het bijbehorende punt binnen de cirkel ligt. Er geldt dan:

$y \leqq \sqrt{1-x^2}$ (immers voor de cirkel geldt $x^2+y^2 = 1$)

Als we nu een groot aantal getallenparen trekken, dan zal de verhouding van de getallenparen die 'OK' zijn en die hieraan niet voldoen bij benadering de waarde $\frac{1}{4}\pi$ opleveren. Dit is de oppervlakte van het cirkelsegment. Het berekenen van deze oppervlakte is een deterministisch probleem dat op deze wijze met behulp van de Monte Carlo steekproeftechniek is opgelost.

III.5.4 Schattingen van de parameters in functionele vergelijkingen

Nadat in de vorige subparagraaf de schatting van parameters van verdelingsfuncties aan de orde geweest is, zal in deze subparagraaf de schatting van parameters in functionele relaties worden behandeld. Functionele relaties zijn relaties die door middel van wiskundige vergelijkingen worden weergegeven.

De eerste stap voor het afleiden van vergelijkingen is verzameling van gegevens betreffende de variabelen in het vergelijkingenstelsel. De volgende stap is na te gaan welk type relatie tussen de variabelen gelegd dient te worden. Hierbij kan voor een eerste indruk gebruik worden gemaakt van een spreidingsdiagram. Als dit tot de conclusie leidt, dat er van een voldoende correlatie sprake is, kan voor het weergeven van de relatie naar een specificatie worden gezocht. Hieronder volgt een aantal mogelijke vergelijkingen als we te doen hebben met een relatie tussen twee variabelen $y = f(x)$.

$Y = a_0 + a_1X$	rechte lijn
$Y = a_1X + a_2X^2$	parabool
$Y = a_0 + a_1X + a_2X^2 + \ldots + a_nX^n$	n°-graadscurve
$Y = a_0 + a_1\log x$	logaritmisch verband
$\log Y = a_0 + a_1\log X$	dubbel log. verband

enzovoorts.

Eén van de meest gebruikte methoden voor het schatten van de parameters in het geval er sprake is van een lineaire vergelijking is de methode van de kleinste kwadraten. Deze methode heeft als uitgangspunt, dat de best passende rechte door de puntenwolk gevormd door de punten

$$(X_1,Y_1),\ (X_2,Y_2),\ (X_3,Y_3),\ \ldots\ldots,\ (X_n,Y_n)$$

de rechte is met de eigenschap:

$$\Sigma e_i^2 = \Sigma(Y_i - \hat{Y}_i)^2 \quad \text{is minimaal.}$$

Hierin is $\hat{Y} = a_0 + a_1 X_i$ en $e_i = Y_i - \hat{Y}_i$

Aan deze eigenschap van de kleinste kwadraten wordt voldaan als a_0 en a_1 zodanig gekozen worden, dat

$$a_0 = \bar{Y} - a_1\bar{X} \qquad \bar{Y} = \frac{\Sigma Y_i}{n}, \quad \bar{X} = \frac{\Sigma X_i}{n}$$

en $$a_1 = \frac{\Sigma x_i y_i}{\Sigma x_i^2} \qquad x_i = X_i - \bar{X}, \quad y_i = Y_i - \bar{Y}$$

Andere, meer ingewikkelde formules zijn af te leiden als de variabele Y van meer variabelen afhankelijk is. Zoals gezegd, mag de methode van de kleinste kwadraten alleen worden toegepast als we te maken hebben met lineaire vergelijkingen. Niet-lineaire vergelijkingen kunnen vaak tot lineaire worden getransformeerd, zodat toch de kleinste kwadratenmethode gebruikt kan worden. Voorbeelden hiervan zijn:

$$Y_t = AB^t v_t \rightarrow \log Y_t = \log A + t \log B + \log v_t$$

$$Y = a + \frac{b}{x} \rightarrow Y = a + bW \text{ met } W = \frac{1}{x}.$$

Voor de uitvoering van de berekeningen kan van computerprogramma's gebruik worden gemaakt, zodat de rekentechnische zijde van de schatting van de parameters geen probleem hoeft op te leveren.

Voor de beoordeling van de kwaliteit van de aanpassing van de regressielijn aan de puntenwolk zijn een aantal maatstaven ontwikkeld:

a. de Von Neumann-ratio
b. de standaardfouten van de regressiecoëfficiënten
c. de correlatiecoëfficiënt.

Meer omvangrijke computerprogramma's voeren naast de parameterschattingen ook berekeningen uit voor deze drie maatstaven. Voor een behandeling van deze onderwerpen verwijzen we naar de appendix.

Als het onmogelijk is de parameterwaarden te schatten met behulp van de methode van de kleinste kwadraten, omdat bijvoorbeeld empirische gegevens ontbreken, zal men zijn toevlucht moeten nemen tot meer subjectieve schattingen. Een traditionele benadering van de besluitvorming is in dit geval het houden van *discussiebijeenkomsten* met een aantal experts. Aan een aantal met het onderwerp vertrouwde deskundigen wordt gevraagd om via een discussie uitspraken te doen die tot invulling van de parameterwaarden kunnen leiden.

Stel bijvoorbeeld dat een bierbrouwerij een relatie wil leggen tussen de bierverkoop en de gemiddelde maandtemperatuur. Door nu erkende bierdrinkers te vragen naar of te laten discussiëren over hun koopgewoonten kan misschien een indicatie omtrent het bedoelde verband worden verkregen.

Het nadeel van deze beslissingsmethodiek kan zijn, dat de groepsmening sterk beïnvloed wordt door dominerende persoonlijkheden. Een methodiek waarbij men deze subjectieve beïnvloeding probeert te voorkomen is de *Delphi-methode*. Bij deze methode mogen de experts onderling niet communiceren. Zij krijgen individueel, in een aantal ronden, vragen voorgelegd. Na de eerste ronde worden de antwoorden op de vragen uit die ronde verwerkt en in de tweede ronde aan de deskundigen voorgelegd. De bedoeling is om op deze manier na een aantal ronden tot één gemeenschappelijke opvatting te komen. Het vraag- en antwoordproces wordt begeleid door een neutrale coördinator.

Aan de methode liggen de veronderstellingen ten grondslag, dat experts op een bepaald terrein ook in staat zijn 'in de toekomst te kijken' en dat de uitkomst van het proces minstens even goed is als de opvatting van elk der individuen.

Hoewel niet mag worden verwacht, dat men langs deze weg een nauwkeurige schatting van de parameters verkrijgt, zal een redelijk betrouwbare 'range' van de schatting reeds aanzienlijke kosten- en tijdbesparingen in de experimenteerfase van het simulatie-onderzoek met zich mee kunnen brengen.

De Delphi-procedure heeft ongetwijfeld nadelen. Zo zullen de schattingen van de parameters met de nodige voorzichtigheid gebruikt moeten worden. Dit nadeel is evenwel relatief, omdat de uitkomsten waarschijnlijk de 'beste' zijn gezien de mogelijkheden.

Aan het slot van deze paragraaf willen we ten behoeve van insiders nog een opmerking maken over verschillen tussen simulatiemodellen en econometrische modellen.

In de eerste plaats worden in econometrische modellen de exogene variabelen veelal als deterministisch beschouwd, terwijl de parameters met behulp van statistische schattingsprocedures worden geschat. Bij simulatie daarentegen veronderstelt men dat de variabelen stochastisch zijn. De variabelen zijn kansvariabelen met een bepaalde kansverdeling c.q. kansdichtheid. De relaties worden in de vorm van hypothesen - die later gevalideerd moeten worden - geformuleerd. De parameterwaarden worden gekozen en de besproken schattingstechnieken zijn bedoeld om de keuzes van de parameters te onderbouwen.

De uitgangspunten tussen simulatie en econometrie zijn dus volkomen verschillend. Met simulatie heeft men een veel flexibeler gereedschap om een model te ontwikkelen. Ook is men bij simulatie niet gebonden aan lineariteit van de functionele relaties, noch behoeven de relaties in een functionele vorm te worden uitgedrukt. Bovendien is het in een simulatiemodel mogelijk met 'if .... then .... else ....' relaties te werken, terwijl tijdsafhankelijke relaties evenmin een probleem vormen.

In dit hoofdstuk zijn nu de bouwstenen voor het maken van een simulatiemodel aangereikt. Als deze bouwstenen in een concrete situatie worden gebruikt, zal de onderzoeker nu over een kwantitatief model beschikken. Dit model moet nog wel worden getoetst. Met een kritische blik, met behulp van experts en van bepaalde technieken zal het model aan een zekere evaluatie onderwerpen moeten worden. Over dit onderwerp gaat de volgende paragraaf.

III.6 EVALUATIE VAN HET MODEL

Modelbouw is een uitdaging voor onderzoekers om onbekende terreinen te exploreren. Door middel van een model proberen we als onderzoekers een onbekend terrein in kaart te brengen. Een groot probleem daarbij is dat het onbekende terrein nooit volledig bekend zal worden. We zullen altijd meer of minder in het duister blijven tasten. Ook al bouwen we in onze eigen ogen, en zelfs in de ogen van anderen nog zo'n goed model, we zullen nooit kunnen zeggen dat het model een juiste afbeelding is van de werkelijkheid.

Immers wat is 'juistheid' van een model? Houdt dit in dat we een model hebben gebouwd waarbij alle systeemvariabelen en relaties worden weergegeven door overeenkomstige modelvariabelen en relaties of houdt dat in dat het model aanvaardbaar nauwkeurige voorspellingen geeft of -----, vult u zelf maar in.

In de literatuur wordt een aantal eisen genoemd waaraan een simulatiemodel dient te voldoen

- het model moet eenvoudig zijn;
- het model moet doelgericht zijn;
- het model moet robuust zijn, d.w.z. het mag geen 'vreemde' uitkomsten geven;
- het model moet overzichtelijk zijn en het moet gemakkelijk te hanteren zijn;
- het model moet op de belangrijke punten volledig zijn;
- het model moet vanuit een eenvoudige opbouw verder te ontwikkelen zijn;
- het model moet gemakkelijk aan te passen zijn.

Moet er nu aan al deze eisen voldaan zijn om te kunnen zeggen dat het model een juiste afbeelding is? Welneen! Het begrip 'juistheid' dient ingevuld te worden afhankelijk van de doelstelling (zie § II.7) die voor het te ontwikkelen model wordt gehanteerd. Eenvoudige modellen kunnen goede voorspellingen geven, zolang voldaan wordt aan de voorwaarden die voor het model gelden, maar deze modellen hoeven bepaald niet robuust te zijn. Bij veranderende omstandigheden waarmee in het model geen rekening is gehouden geeft zo'n eenvoudig model 'vreemde' uitkomsten. Als de gebruiker van het model de beperkingen nu maar inziet en op tijd de veranderingen onderkent, is dit niet ernstig.

Het is in deze fase van het onderzoek overigens onmogelijk om het geconstrueerde model volledig te evalueren. We kunnen dit nog slechts in een zeer beperkte mate doen. In deze fase van het onderzoek dient het model te worden geëvalueerd op:

de juistheid van de afbeelding, in de zin van 'justificatie'.

Hierbij dient men zich af te vragen of het model, gegeven de doelstelling een adequate afbeelding is van het concrete systeem.

De term justificatie geeft hier wellicht het beste weer wat wordt bedoeld. Deze term wijst op: verantwoording en rechtvaardiging. De onderzoeker legt verantwoording af aan zichzelf èn aan zijn opdrachtgevers, hij rechtvaardigt zijn beslissingen t.a.v. de keuze van de variabelen, van de relaties en van de parameters.

Er kunnen bij deze justificatie ook experts worden ingeschakeld. Het model wordt dan voorgelegd aan deskundigen met de vraag in hoeverre zij het model geloofwaardig vinden, gegeven de doelstellingen van het onderzoek. Deze vraag kan nader worden uitgewerkt en wel als volgt:

- zijn de opgenomen variabelen van voldoende gewicht voor de verklaring van de endogene variabele(n)?
- is er met alle relevante exogene variabelen rekening gehouden in het model?
- zijn de functionele verbanden tussen de afhankelijke en de onafhankelijke variabelen op de juiste wijze gespecificeerd?
- zijn de schattingen van de parameters reëel?

Men spreekt in dit verband ook wel van 'face-validity'.

Als op grond van deze evaluatie het model wordt 'vrijgegeven' kan worden overgegaan tot de volgende fase van het onderzoek. Mocht dit niet het geval zijn, dan moet worden teruggekoppeld naar één van de voorgaande fasen.

Tenslotte willen we hier nog verwijzen naar hoofdstuk VI dat in bredere zin het onderwerp validatie behandelt.

SLOTOPMERKING

We beschikken nu over een 'model'. Dit betekent echter niet dat we klaar zijn. Als namelijk de relaties in een kwantitatieve vorm (bijvoorbeeld in de vorm van een computerprogramma) zijn uitgeschreven, beschikt men nog niet over een verklaring! Dit is een fout, die veel gemaakt wordt. Het bouwen van een model is geen doel, maar een middel. Het model moet nog opgelost worden. Simulatie is een heuristische oplossingsmethode, waaruit een verzameling alternatieve oplossingen tevoorschijn moet komen.

III.7 OPGAVEN

Opgave 1

Een veel gebruikte formule voor het verkrijgen van een reeks toevalsgetallen is:

$$x_{i+1} = x_i * a \text{(modulo m)}$$

De startwaarde voor de randomgenerator is x_0.

a. Stel dat $x_0 = 3$, $a = 7$ en $m = 15$.
Genereer met de hand een tiental toevalsgetallen.

Om m.b.v. een computer de reeks te doen genereren kunnen aan a en m enige eisen worden gesteld, waarmee de kwaliteit van de randomgenerator samenhangt. Zo'n randomgenerator is dan machine-afhankelijk. Uit de literatuur volgen deze aanbevelingen:

x_0 = oneven

$a = 8k \pm 3$ (k is een niet-negatief geheel getal)

Kies voor m het grootste gehele getal dat een geheugenwoord kan bevatten. Bij een '32-bitsmachine' is $m = 2^{31}-1$ of 2^{31} (macht van 2).
Het getal a is vaak 16807.

b. Schrijf een procedure in Pascal, Simula, BASIC of een andere beschikbare taal om randomgetallen te produceren.

c. Voldoen de sub b. geproduceerde getallen aan een uniforme verdeling? Een chi-kwadraat toets is als toetsingsmiddel goed bruikbaar (zie Appendix A).

d. Tot welke bezwaren kan de machine-afhankelijkheid van een randomgenerator leiden?

e. Toets uw procedure aan de lijst van eisen die aan het slot van § III.5.1 zijn genoemd.

Opgave 2

Een stichting Studentenhuisvesting beheert zeven studentenflats. Deze flats zijn in twee blokken te onderscheiden. In drie flats (blok I) met in totaal 1000 kamers is het verloop aanmerkelijk groter dan in de vier andere flats (blok II), met eveneens 1000 kamers.
Aanmelding voor deze kamers vindt vooral plaats in de maanden mei tot en met juli. Daarna lopen de aanmeldingen terug. De kamers worden in volgorde van aanmelding toegewezen. Als de flats vol zijn worden de belanghebbende studenten op een wachtlijst gezet.

a. Formuleer zelf een doelstelling voor een simulatie-onderzoek en geef globaal aan langs welke weg u dit onderzoek zou willen uitvoeren.

b. Formuleer het probleem met behulp van de diverse soorten van variabelen, relaties en parameters.

c. Geef aan op welke manier u aan de noodzakelijke gegevens denkt te komen om het model 'rond' te maken zodanig dat u tot het experimenteel ontwerp (de volgende fase van het onderzoek) over kunt gaan.

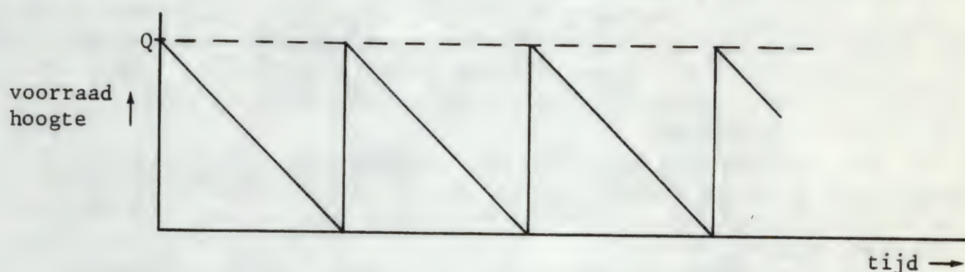
d. Welke methoden kunt u hanteren om tot een evaluatie van het ontworpen model te komen?

Opgave 3

Bedenk zelf een concreet probleem waarvan u denkt dat dit geschikt is om door simulatie te worden onderzocht.
Beantwoord dezelfde vragen als die bij opgave 2 zijn gesteld.

Opgave 4

Ontwerp een simulatiemodel van een voorraadsysteem dat door de volgende schets wordt gekarakteriseerd. Beantwoord daarbij ook de vragen uit opgave 2.



Opgave 5

Schrijf in Pascal, Simula, BASIC of een andere door u gebruikte taal procedures om toevalsgetallen te trekken uit:

a. de negatief exponentiële verdeling;
b. de normale verdeling;
c. de Poisson verdeling.

HOOFDSTUK IV

Het Opzetten van Simulatie-Experimenten

IV.1 INLEIDING

In figuur III.2c is een schets gegeven van de fasering van simulatie-onderzoek. Volgens deze schets volgt na de modelontwikkeling het experimenteel ontwerp (Engels: experimental design), ook wel aangeduid als de keuze van de proefopzet. Deze fase is bedoeld om na te gaan 'welke experimenten nodig zijn om aan de gevraagde informatie te komen' en 'hoe deze experimenten zo efficiënt mogelijk uitgevoerd kunnen worden'. In § 2 van dit hoofdstuk gaan we heel globaal op deze vragen in. Aan de hand van enkele voorbeelden wordt uiteengezet wat experimenteel ontwerp is. Verder wordt de plaats van het experimenteel ontwerp in het totale simulatie-onderzoek besproken.

Om het doel van het experimenteel ontwerp zoveel mogelijk te realiseren is enige structurering van het ontwerpproces wenselijk. In § 3 bespreken we dit proces; we lichten het toe aan de hand van 'de haringkar'.

Het ontwerpproces kent *twee hoofdfasen*, die van de *strategische* en die van de *tactische planning*. Bij de strategische planning gaat het vooral om vragen als:

- welke exogene variabelen gebruiken we om het effect hiervan te onderzoeken op de endogene variabelen;
- welke waarden geven we de exogene variabelen?
- welke belangrijke beslissingen moeten worden genomen in het reële systeem? Door te experimenteren willen we het effect van die beslissingen te weten komen.

De strategische planning wordt toegelicht in § 4.

Bij de tactische planning gaat het om de invulling van de experimenten met het oog op een zo efficiënt mogelijke aanpak. Voorbeelden van tactische planningsvraagstukken zijn:

- de omvang van de steekproef;
- de startwaarden van de toestandsvariabelen;
- het aantal herhalingen van een experiment.

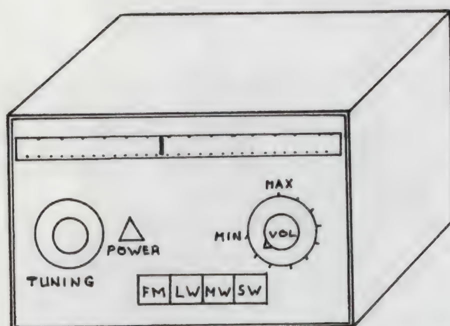
Terwijl het bij de strategische planning gaat om de doelstellingen en het opstellen van een plan om deze doelstellingen te realiseren, gaat het bij de tactische planning om de realisering van het plan tegen zo laag mogelijke kosten.

In de §§ 5-9 worden verschillende vraagstukken die tot de tactische planning behoren meer gedetailleerd behandeld. Een volledige behandeling is in het kader van dit boek niet goed mogelijk. Daarvoor is teveel kennis van statistiek vereist. In appendix A geven we een overzicht van statistische technieken die voor de tactische planning van belang kunnen zijn.

Zoals gebruikelijk wordt het hoofdstuk afgesloten met een paragraaf opgaven.

IV.2 HET EXPERIMENTEEL ONTWERP IN GROTE LIJNEN

Ter inleiding op de vraag: 'Wat is experimenteel ontwerp?' bespreken we een voorbeeld. We nemen een eenvoudig radiotoestel met daaraan de volgende knoppen: aan/uit, hard/zacht met 10 verschillende standen, golflengte (FM, LW, MW, SW) en een afstem-knop met per band: 10, 10, 40 respectievelijk 5 te ontvangen stations met een acceptabele ontvangstkwaliteit.



Laten we aannemen, dat we op een systematische wijze met het toestel gaan experimenteren. Dit kan op een groot aantal manieren gebeuren.

Vraag: Hoeveel verschillende varianten van verschillende geluids-kwaliteit zijn, als de radio aanstaat, mogelijk?

Het is onwaarschijnlijk, dat u alle varianten in de praktijk zult gaan onderzoeken. Al heel snel zal bij daadwerkelijk gebruik van het toestel blijken, dat een tiental varianten voor u het meest aantrekkelijk zijn. Op die varianten zult u zich bij het instellen van het radiotoestel dan ook vooral concentreren.

Eenzelfde redenering gaat op voor experimenten met een simulatiemodel. Stel, dat een levensmiddelengroothandel een oplos-sing zoekt voor de vestigingsplaatsen van een magazijn ter bevoor-rading van de detaillisten. De volgende factoren spelen bij deze beslissing een rol:

- vestigingsplaats (10 mogelijkheden)
- vorm van bedrijfsvoering (3 mogelijkheden)
- transportmethode (5 mogelijkheden).

Evenals bij de radio zijn een groot aantal varianten mogelijk.

Vraag: Hoe groot is het aantal mogelijke varianten?

Bij elk van de varianten zal men ook te maken hebben met kansvariabelen, zoals bijvoorbeeld bestelfrequentie en bestelgrootte. Teneinde de invloed van het stochastische element van de 'random generator' weg te werken, zal eenzelfde variant meerdere malen worden herhaald. Hierdoor neemt het aantal malen dat men een experiment uitvoert nog eens toe.

Vandaar dat wordt geprobeerd het aantal varianten krachtig te beperken. Bekeken moet dan worden welke combinaties de meeste informatie zullen opleveren. Met andere woorden, de economische noodzaak dwingt ons om na te gaan op welke wijze de meeste informatie tegen de minste kosten verkregen kan worden.

Als belangrijke kostenfactoren bij simulatie kunnen worden genoemd mankracht, tijd en computerfaciliteiten.

Mankracht: vaak is er sprake van een onderzoekteam van 2-5 leden, die van een behoorlijk niveau moeten zijn om het onderzoek te kunnen uitvoeren.

Tijd: de benodigde tijd voor een onderzoek is vaak moeilijk te schatten; toch is beheersing van het tijdsaspect bij veel onderzoeken zeer noodzakelijk.

Computerfaciliteiten: voor simulatie-onderzoek is een computer noodzakelijk; het schatten van de capaciteit en zeker van de computertijd is lastig, maar moet wel gebeuren.

Men zal in de experimentele fase van het project moeten nagaan wat de nog beschikbare middelen zijn, uitgedrukt in de drie genoemde grootheden. Op deze gegevens, die men gewoonlijk als randvoorwaarden beschouwt, zal het experimentele ontwerpplan gebaseerd moeten worden.

In samenhang met de vaststelling van de beschikbare middelen is de vraag van de gewenste informatie van belang. Iedere onderzoeker zal zoveel mogelijk informatie (≡ eindprodukt) willen krijgen tegen zo laag mogelijke kosten. Welke experimenten uitgevoerd moeten worden om de gewenste informatie te verkrijgen, dat wil zeggen van welke varianten bij voorbaat vastgesteld wordt dat ze plausibele mogelijkheden zijn om onderzocht te worden, is de belangrijkste vraag die bij de strategische planning aan de orde komt. In § 4 van dit hoofdstuk zal de strategische planning worden behandeld.

In dit verband speelt ook een rol het feit, dat de gegevens van de experimenten verwerkt moeten worden. Elk experiment levert een bepaalde hoeveelheid gegevens. Om aan deze gegevens relevante en betrouwbare informatie te onttrekken, zullen bewerkingen nodig zijn. De doelstellingen van de simulatie bepalen welke bewerkingen dat zijn. meestal worden deze bewerkingen uitgevoerd met behulp van statistische technieken.

Tot de meest voor de hand liggende bewerkingen behoren:

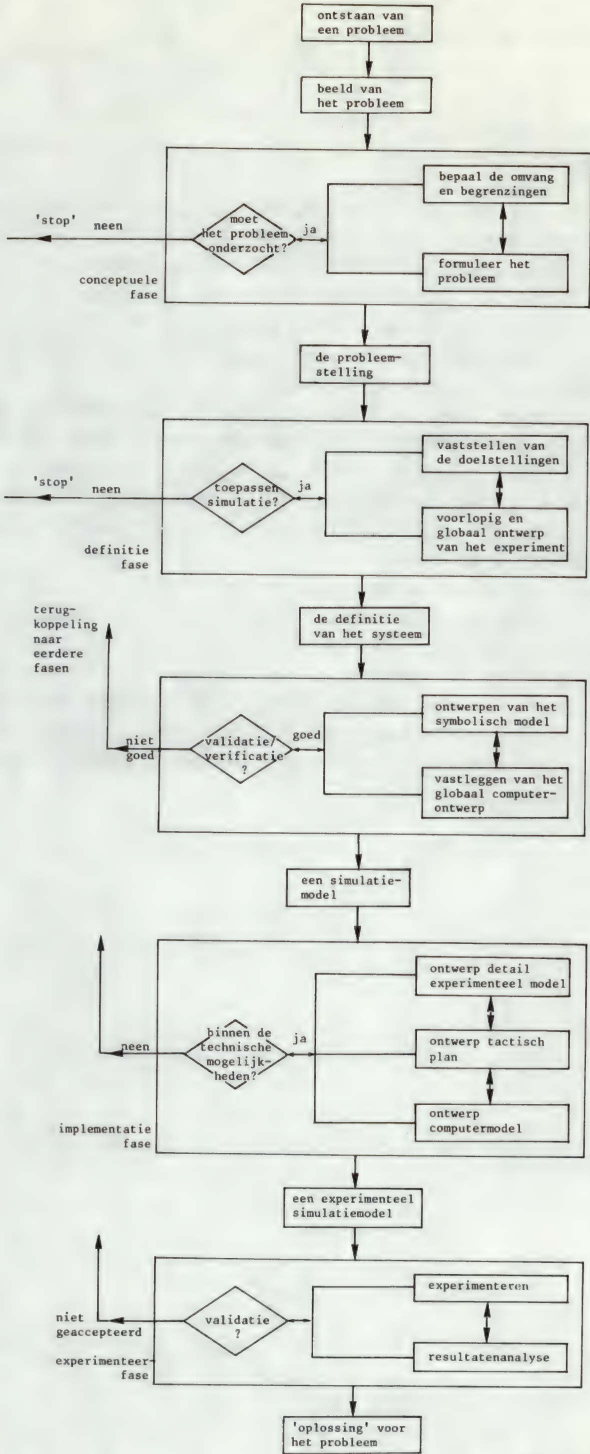
- het vergelijken van alternatieven door middel van gemiddelden en varianties;
- het bepalen van de invloed van verschillende variabelen en het interval waarin zij van belang zijn;
- het bepalen van de onderlinge afhankelijkheid der variabelen;
- het zoeken naar optimale waarden van een variabelenverzameling.

Het simulatiemodel is een experimenteel instrument in handen van de onderzoeker, op soortgelijke wijze als bij laboratoriumexperimenten. Daarom zullen ook soortgelijke overwegingen een rol spelen bij het construeren van de opzet van het simulatie-experiment. Hierbij kan als voorbeeld aangehaald worden het probleem dat bij veel economisch onderzoek naar voren komt, namelijk het van elkaar afhankelijk zijn van een aantal modelvariabelen. Om dit probleem op te lossen zijn doorgaans grote aantallen experimenten nodig. In dit soort gevallen is dan een goede opzet van groot belang.

Het belang van een zorgvuldig experimenteel ontwerp wordt ook nog benadrukt door te bedenken dat de geldigheid ('validity') van het simulatiemodel ervan afhankelijk is. De geldigheid is immers altijd relatief en wel binnen de grenzen opgelegd door het experimentele kader. Dit kader legt namelijk de grenzen vast van de mogelijke waarnemingen aan het model. Een model zou door bepaalde experimenten geldig verklaard worden, terwijl andere experimenten juist het tegendeel aantonen.

Het experimenteel ontwerp komt bij een simulatie-onderzoek verschillende keren aan de orde. Dit blijkt niet zonder meer uit figuur III.2c, hoewel wij ter toelichting van deze figuur wel hebben gewezen op een zekere parallelle uitvoering van de verschillende fasen. Om de belangrijkheid van het experimenteel ontwerp in het gehele onderzoek tot uitdrukking te brengen, hebben we figuur IV.2a opgenomen. Bij bestudering hiervan zien we, dat de vraag 'Welke experimenten gaan we uitvoeren en hoe?'verschillende keren wordt gesteld. Behalve in de experimentele ontwerpfase zelf moet er ook reeds in de probleemformuleringsfase aandacht aan het experimenteel ontwerp worden besteed. Wanneer de probleemformulering 'voltooid' is en besloten is dat men computer-simulatie gaat toepassen, zullen ook de doelstellingen bekend en geformuleerd zijn. De opzet van het experiment hangt ten nauwste samen met de doelstellingen. Daarom zetten we op dit moment tevens *een voorlopig experimenteel plan* op. Samen met de doelstelling is namelijk bekend wat de onderzoeker wenst te weten van het systeem. Het experimenteel

Het experimenteel plan zal de vraag: 'Hoe denkt men aan die wetenschap te komen?', moeten beantwoorden. Het experimenteel plan zal zelfs al betrekkelijk gedetailleerd moeten zijn, omdat het te construeren model beter en efficiënter ontworpen kan worden


ontstaan van een probleem
beeld van het probleem
'stop'
neen
moet het probleem onderzocht?
ja
bepaal de omvang en begrenzingen
formuleer het probleem
conceptuele fase
de probleem-stelling
'stop'
neen
toepassen simulatie?
ja
vaststellen van de doelstellingen
voorlopig en globaal ontwerp van het experiment
definitie fase
terug-koppeling naar eerdere fasen
de definitie van het systeem
niet goed
validatie/verificatie?
goed
ontwerpen van het symbolisch model
vastleggen van het globaal computer-ontwerp
een simulatie-model
neen
binnen de technische mogelijk-heden?
ja
ontwerp detail experimenteel model
ontwerp tactisch plan
ontwerp computermodel
implementatie fase
een experimenteel simulatiemodel
niet geaccepteerd
validatie?
experimenteren
resultatenanalyse
experimenteer-fase
'oplossing' voor het probleem

Figuur IV.2a

worden om straks de benodigde experimentele gegevens te kunnen genereren, als men al enig idee heeft hoe de experimenten eruit gaan zien.

Tevens moet niet uit het oog worden verloren dat in het computermodel een goed gedeelte van het experimenteel ontwerp moet zijn ingebouwd.

Ten derde zal men de analyse van de uitvoergegevens bij voorkeur ook automatiseren. Ook deze 'outputanalyse' zal in het computermodel ingebouwd moeten worden. Redenen te over om niet pas nadat men het simulatiemodel heeft ontworpen en gecodeerd zich met de vraag omtrent het experimenteren bezig te houden en dan tot de ontdekking te komen dat de gehele programmaconstructie en uitvoerformulering moeten worden gewijzigd.

Is de noodzaak voor het voorlopige experimentele plan wel aangetoond, hoe de concrete uitwerking ervan moet plaatsvinden is een verre van eenvoudige zaak. Men weet immers van het te onderzoeken systeem nog bar weinig (anders zou het onderzoek wellicht geen doorgang behoeven), derhalve is ook nog niet duidelijk hoe het model eruit komt te zien.

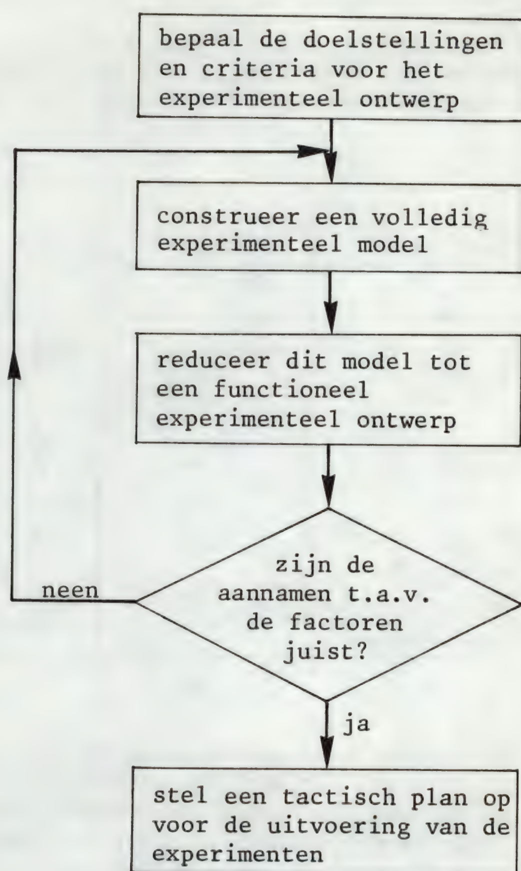
Herhaalde terugkoppeling naar deze fase, terwijl de analist voortschrijdt in zijn kennis en inzicht, is daarom een vereiste.

Nadat het model geconstrueerd en gevalideerd is, het computerprogramma gereed en uitgetest, komt het punt waarop *het echte experiment* ingaat. Dit is simulatie in de beperkte betekenis van het woord. Om het 'echte experiment' te doen slagen, zal het gedetailleerde experimenteel ontwerp in het computerprogramma ingebouwd moeten zijn.

IV.3 HET (EXPERIMENTEEL) ONTWERPPROCES

Elk ontwerp en dus ook het ontwerp van simulatie-experimenten moet op een systematische manier aangepakt worden. Daarom stellen we een benaderingswijze voor in de vorm van een gefaseerd proces, waarvan het resultaat een ontwerp is van de experimenten die we willen gaan uitvoeren.

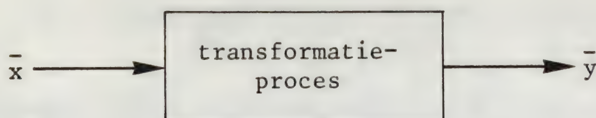
In het kort is dit proces als volgt te schematiseren:



Figuur IV.3a

We zullen dit proces illustreren met een bekend voorbeeld, dat van de haringkar. Eerst echter bespreken we een aantal termen.

Wanneer we een simulatiemodel beschouwen als een 'input-output'-model, ook wel een 'black box'-model genoemd, dan zijn er twee grootheden van belang, namelijk x en y, zie figuur IV.3b.



Figuur IV.3b

$\bar{x}$ en $\bar{y}$ zijn symbolen voor de vector van invoervariabelen respectievelijk van uitvoervariabelen.

Er zijn nogal wat benamingen voor de x en y vectoren, afkomstig uit verschillende disciplines die op de een of andere wijze in hun research met modellen werken, zoals biologie, sociologie, enzovoorts.

$\bar{x}$		$\bar{y}$	
Engels	Nederlands	Engels	Nederlands
factor	factor	response	uitkomst
input variable	invoervariabele	output variable	uitvoervariabele
exogenous variable	exogene variabele	endogenous variable	endogene variabele
treatment		yield	
independent variable	onafhankelijke variabele	dependent variable	afhankelijke variabele
	omgevingsvariabele		uitgangsvariabele
	beslissingsvariabele		te verklaren variabele
	regel variabele		
	instrumentele variabele		
	verklarende variabele		

Wij zullen in het kader van het experimenteel ontwerp de termen 'factor' en 'response' of 'responsevariabele'*) gebruiken.

Eén vraag staat nu centraal, namelijk: 'Welk effect heeft een verandering in $\bar{x}$ op $\bar{y}$?' Het experimenteel ontwerp bestaat daaruit dat we een lijst van factoren samenstellen, die we al experimenterend willen veranderen om te zien hoe y reageert. Daarbij behoort ook de vaststelling welke waarden die factoren moeten hebben (Engels: 'levels').

Het model van het experimenteel ontwerp dat op deze wijze verkregen wordt, is een ideaal model; d.w.z. met economische en

*) We schrijven verder response of responsevariabele, dus zonder het gebruik van aanhalingstekens.

statistische beperkingen wordt nu geen rekening gehouden. We noemen dit ook wel het structurele model. Let wel: dit is niet hetzelfde als het simulatiemodel waarmee we experimenteren. Ga dit zorgvuldig na.

Het uiteindelijke ontwerp van het experimenteel model wordt vastgelegd door de opgelegde criteria. Deze zullen eerst moeten worden omgeschreven. De volgende vragen moeten worden gesteld en beantwoord.

1. Welke zijn de factoren, waarmee we willen experimenteren en waarvan we het effect op het systeem willen bepalen?

2. Van welke factoren moeten de aard en het aantal waarden dat we willen gebruiken, worden vastgesteld?
   - Zijn de waarden van de betreffende factor kwalitatief of kwantitatief?
   - Zijn de waarden vast (fixed) of random?
   - Moeten er effecten gemeten worden, die niet lineair zijn?
   - Krijgen alle factoren een gelijk aantal waarden?
   - Zijn de waarden van de factoren waarneembaar?

3. Het aantal metingen van de responsevariabelen.
   - Kunnen interactie-effecten tussen de factoren waargenomen worden?
   - Welke nauwkeurigheid wordt er vereist?

VOORBEELD

In hoofdstuk I is uitvoerig ingegaan op de haringkar. Een experimenteel ontwerp voor de haringkar zullen we nu volgens het schema van figuur IV.3a opzetten.

1. Doelstellingen en criteria.

Het ging erom te onderzoeken op welke wijze de wachttijden van de klanten aan de haringkar kunnen worden teruggebracht. We simuleren het systeem bij dagen met een normaal gemiddeld klantenaankomstpatroon en bij drukke dagen. We vergelijken de wachttijden die er ontstaan, wanneer er een of twee bedienden de klanten helpen. Er worden voorwaarden gesteld aan de duur van de wachttijd (maximaal 5 minuten) en de lengte van de wachtrij. De simulatie experimenten moeten ons inzicht verschaffen over de invloed van het aantal bedienden (een beslissingsvariabele) op het gedrag van het systeem: de wachttijd van de klanten. Deze wachttijd is dus de responsevariabele.

2. Een volledig experimenteel model (zie ook § IV.5).

Alle factoren die van invloed zijn op de responsevariabele wachttijd, zullen onmogelijk meegenomen kunnen worden. De scherpte

of botheid van het fileermes dat voor het schoonmaken van de vis wordt gebruikt, zou wel eens een rol van betekenis kunnen spelen. Het antwoord op de vraag van de relevante factoren zal dan ook voornamelijk door de opdrachtgever, die kennis heeft van het reële systeem, gegeven moeten worden. Wij stellen de volgende factoren vast:

x1: aankomstpatroon van de klanten
x2: de boodschappenlijst van de klanten
x3: de dag in de week
x4: het uur van de dag
x5: het seizoen
x6: de rijdiscipline
x7: het aantal bedienden.

Deze factoren hebben uiteenlopende waarden, ze zijn niet alle onafhankelijk van elkaar en tenslotte zullen ze elk niet een even sterke invloed uitoefenen op het gedrag van het systeem. Daarom zullen we aan de hand van een aantal veronderstellingen de factoren terugbrengen tot een verantwoord aantal. Deze veronderstellingen moeten natuurlijk wel op de een of andere manier op hun juistheid worden getoetst.

3. Het functionele experimenteel model (zie ook § IV.4)

Zoals gezegd brengen we nu het aantal factoren tot een, gezien de doelstellingen van het onderzoek verantwoord aantal terug. We komen dan tot het volgende overzicht.

x1: de aankomsttijd is een factor die zich gedraagt volgens een Poissonproces, we experimenteren niet met andere processen;

x2: deze factor wordt weergegeven door een kansvariabele helptijd met een uniforme verdeling;

x3: een factor met 2 waarden, namelijk gewone dagen en de vrijdag; het verschil is, dat de parameter van de aankomstverdeling anders wordt;

x4: hoewel het aannemelijk is dat de dagen piekuren vertonen, brengen we dit aspect niet in: de factor vervalt;

x5: eenzelfde redenering als voor x4 geldt, ook deze factor vervalt;

x6: de klanten worden geholpen volgens het principe 'first in, first out';

x7: het aantal bedienden is één of twee.

We houden dus twee beïnvloedbare factoren over, namelijk:

x3 met twee waarden: normale dag en vrijdag
x7 met twee waarden: één en twee.

Het functionele experimenteel ontwerp kan als volgt in een tabel worden weergegeven:

x3 \ x7	1	2
normale dag	*	*
vrijdag	*	*

Figuur IV.4b Functioneel experimenteel ontwerp

Op grond van dit model kunnen nu vier experimenten worden uitgevoerd, die bij uitvoering elke informatie zullen opleveren over de responsevariabelen, zoals de wachttijd. Over de vraag hoe experimenten het beste uitgevoerd kunnen worden, gaan de §§ 5-8 van dit hoofdstuk. In § 9 komen we nog eens terug op dit voorbeeld.

IV.4 HET EXPERIMENTEEL MODEL (strategische planning)

In de vorige paragraaf hebben we het experimenteel model onderscheiden in: volledig en functioneel. Het *volledige model* wordt beschreven door alle factoren en alle mogelijke waarden van deze factoren. Als in een bepaald onderzoek k het aantal factoren is en q_i het aantal waarden van de factor i dan kan het aantal experimenten N_s worden berekend uit:

$$N_s = \prod_{i=1}^{k} q_i \qquad i = 1,2,3,\dots,k$$

Als bijvoorbeeld in een bepaald onderzoek $k = 4$ en $q_1 = 2$, $q_2 = 3$, $q_3 = 1$ en $q_4 = 2$, dan is het aantal mogelijke experimenten:

$$N_s = 2 \times 3 \times 1 \times 2 = 12$$

Elk experiment is een bouwsteen of een structurele eenheid van het totale onderzoek. Vandaar dat ook wel de uitdrukking *structureel model* wordt gebruikt.

Het *functionele model* is het model dat ten opzichte van het structurele model beperkt is, omdat het in verband met beperkte middelen niet mogelijk is het structurele model volledig uit te voeren. Zoals uit de bovenstaande formule blijkt, neemt het aantal experimenten van een simulatie-onderzoek al vrij snel toe bij hogere waarden van k en q. Als bijvoorbeeld $k = 6$ en $q = 3$ voor elke factor, dan hebben we al te maken met

$$N_s = 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 3^6 = 279 \quad .$$

Door de grote hoeveelheid waarnemingen die men dan verkrijgt, zal de analyse erg moeilijk zijn. Naast de beperkte middelen is dit ook een reden om het aantal experimenten zoveel mogelijk terug te brengen.

Het functionele experimenteel model wordt beschreven door het aantal experimenten N_f. In de praktijk zal $N_f < N_s$.

We zullen in het vervolg van deze paragraaf achtereenvolgens nader ingaan op de factoren, de waarden van de factoren en op de mogelijkheid het aantal experimenten te beperken.

IV.4.1 De factoren

Een factor is een exogene variabele of een parameter uit het simulatiemodel. Met welke factoren we te maken hebben, is eerst en vooral afhankelijk van de doelstellingen van het onderzoek. Ze zijn vastgesteld in de ontwikkelingsfase van het simulatiemodel. We maken eerst een uitgebreide lijst van alle factoren die van invloed

kunnen zijn op de responsevariabelen. Daarna proberen we deze lijst te reduceren door de meest belangrijke factoren te selecteren. Hoe minder we van het systeem afweten, hoe lastiger dat is. Wel weten we dat het merendeel van de systemen onderworpen is aan het *Pareto principe*, d.w.z. dat slechts weinig factoren significant van invloed zijn op de responsevariabelen. Een vuistregel is, dat ca. 20% van de factoren ca. 80% van het totale effect voor zijn rekening neemt. De hamvraag is dus eigenlijk: welke zijn die weinige, doch belangrijke factoren?

De volgens het onderzoek in aanmerking komende factoren worden geclassificeerd in een drietal categorieën.

1. De variabele kan constant gehouden worden en maakt dan deel uit van de grenzen en condities van het systeem. Deze klasse variabelen behoort niet tot de factoren in het experimenteel model.

2. De factoren die niet controleerbaar of te beheersen zijn. Deze maken deel uit van het toeval, geïntroduceerd via de gebruikte 'random'-generatoren. Een voorbeeld hiervan is het weer. Toch kunnen weercondities wel worden waargenomen en als zodanig kan het weer, wanneer dat van belang is, een factor vormen, zij het een onbestuurbare.



---niet bestuurbaar, wel waarneembaar---

3. De factoren, die meetbaar en beheersbaar zijn, zijn degene waarin we geïnteresseerd zijn in het structureel model. Niet alle factoren die beheersbaar of bestuurbaar zijn, zullen worden meegenomen. In dat geval verdwijnt ook hun effect in de variatie van de toevalscijfers. Een voorbeeld hiervan is 'de dag van de week' (een 'maandag-auto' is berucht).

De factoren, welke men in ieder geval moet introduceren, zijn die waarin men uit hoofde van de doelstelling van het onderzoek geïnteresseerd is. Let wel: deze factoren van 'basis-interesse' zijn niet noodzakelijk diegene, die het grootste effect op de uitkomst hebben.

Hieruit volgt ook een argument voor vroegtijdig experimenteel onderzoek: weet men ten tijde van de constructie van het simulatiemodel nog niet welke factoren later in het experimenteel model opgenomen moeten worden, dan kan dat als consequentie met zich meebrengen dat men het simulatiemodel opnieuw zal moeten ontwerpen. Een andere reden om meer factoren in het model te brengen is om de nauwkeurigheid van de resultaten te vergroten. Ook dit is bij de probleemformulering vastgesteld.

IV.4.2 Het aantal waarden per factor

We stellen eerst vast dat de volgende vragen beantwoord moeten worden:

1. Zijn de waarden van de factor kwalitatief of kwantitatief?
2. Zijn de waarden vast of stochastisch?
3. Zijn er niet-lineaire effecten?
4. Zijn alle factoren op een gelijk aantal waarden te variëren?

Ad 1

Een factor is kwantitatief als de waarden getallen zijn. De variabelen kunnen op de interval- of ratioschaal gemeten worden. Voorbeeld: de weersconditie is kwantitatief: de temperatuur, vochtigheidsgraad in %. Andere voorbeelden zijn kosten, tijd, snelheid. Voorbeelden van kwalitatieve factoren zijn: machines, operators, dag van de week, geografische gebieden, beslissingsregels, rijdisciplines, enzovoorts. Men kan er wel getallen aan verbinden, maar het zijn geen 'echte functies';

de output : $c \cdot \frac{\text{machinenummer}}{\sqrt{\text{operatornummer}}}$ is een onzinrelatie.

Een ander voorbeeld is de haringkar. Het aantal bedienden, als ook de parameters λ en μ van de aankomst- en helptijdverdelingen zijn kwantitatief. De rijdiscipline die geldt, namelijk wie het eerst komt, wordt het eerst geholpen (FIFO) is een kwalitatieve factor.

Ad 2

We spreken van een vaste factor ('fixed factor') als de waarde van de variabele 'vast' is gekozen. Zo kan bijvoorbeeld bij een experiment van een investeringsmodel voor een waarde van 11% voor de interestvoet worden gekozen. Bij een volgend experiment kan dan met 11,5% of 12% worden gewerkt. Als alle exogene variabelen op bepaalde waarden worden vastgelegd, is het stochastische element verdwenen. Laten we daarentegen de waarden van één of meer factoren door het toeval bepalen, dan is er sprake van een stochastisch experiment. Vaak zullen in een experiment beide categorieën van factoren voorkomen.

Ad 3

Het minimum aantal waarden is twee, daar er anders geen sprake meer is van een factor. Het aantal waarden moet niet groter worden dan noodzakelijk is om het doel van het experiment te bereiken. Dat bepaalt de bovengrens. Elke waarde méér kost alleen maar geld. Wanneer men besluit tot twee waarden voor een factor, dan impliceert dit dat het effect wordt beschouwd lineair te zijn.

Verwacht de onderzoeker echter kwadratische effecten, dan moet hij minstens drie waarden instellen, enzovoorts. Dit geldt nadrukkelijk voor kwantitatieve factoren.

Ad 4

Wanneer men alle factoren evenveel waarden laat aannemen, spreekt men van een symmetrisch model: $N = q^k$.

Dit kan voordelen hebben bij het analytisch onderzoek van de uitkomsten, zoals bijvoorbeeld bij Variantie-analyse. Het loont dus de moeite te onderzoeken of dit mogelijk is.

IV.4.3 Het functionele experimenteel model

Aan het begin van deze paragraaf is onderscheid gemaakt tussen het structurele en het functionele experimenteel model. Het functionele model is beperkt ten opzichte van het structurele model, doordat het aantal experimenten wordt ingeperkt. We schreven reeds, dat het doel van het functionele model is het aantal experimenten zodanig te beperken, dat voldoende informatie gezien het doel van het onderzoek kan worden verkregen.

Het is helaas niet mogelijk om exacte regels te geven voor de manier waarop het aantal experimenten kan worden beperkt. Er zijn in de literatuur van het experimenteel ontwerp, waarvan de theorie is ontwikkeld vanaf de twintiger jaren en waarvan toepassingen bestaan in de landbouw, de chemie, enzovoorts, wel bepaalde mogelijkheden aangegeven, maar deze staan voor wat betreft de toepassing op vraagstukken uit de sociale wetenschappen nog in de kinderschoenen. Om het aantal experimenten te beperken maken we gebruik van de statistische proefopzet*).

Veronderstel dat in een experimenteel model drie factoren x_1, x_2 en x_3 relevant zijn. Elk van de factoren heeft 2 waarden, die we aangeven met respectievelijk +1 en -1. (Een 'hoge' en een 'lage' waarde; 1 en -1 zijn symbolische waarden.) Wanneer we alle combinaties willen bestuderen, zijn $2^3 = 8$ experimenten nodig. Als we evenwel veronderstellen, dat het verband tussen de responsevariabele y en de drie factoren door de volgende geschatte regressievergelijking wordt weergegeven:

---

*) Zie: J.P.C. Kleijnen, The role of statistical methodology in simulation, in: Methodology in systems modeling and simulation, Amsterdam, 1979.

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_3 \qquad \text{(vergelijking IV.1)}$$

dan kunnen we de parameters b_i $(i = 0,1,2,3)$ schatten uit slechts vier experimenten. De parameters in deze vergelijking kunnen worden aangemerkt als de effecten van de factoren op de responsevariabele.

Deze vier experimenten kunnen bijvoorbeeld worden uitgevoerd zoals weergegeven is in tabel IV.4a.

Experiment	x_1	x_2	x_3 $(= x_1x_2)$
1	1	1	1
2	-1	1	-1
3	1	-1	-1
4	-1	-1	1

Tabel IV.4a Experimenteel model voor drie factoren

De combinaties worden als volgt bepaald. Schrijf eerst alle mogelijke combinaties voor x_1 en x_2 uit $(2^2 = 4)$ en voeg dan de waarde van x_3 toe door de waarden van x_1 en x_2 met elkaar te vermenigvuldigen. In dit voorbeeld is sprake van een zogenaamde 2^{3-1} proefopzet, omdat er drie factoren zijn waarbij de waarde van de derde factor afhankelijk wordt gesteld van de twee andere factoren.

Op dezelfde wijze als voor drie factoren kan voor bijvoorbeeld zeven factoren het aantal experimenten worden gereduceerd. Indien een 2^{7-4} proefopzet wordt gevolgd, kan tabel IV.4b worden opgesteld.

Experiment	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	-	-	-	+	+	+	-
2	+	-	-	-	-	+	+
3	-	+	-	-	+	-	+
4	+	+	-	+	-	-	-
5	-	-	+	+	-	-	+
6	+	-	+	-	+	-	-
7	-	+	+	-	-	+	-
8	+	+	+	+	+	+	+

Tabel IV.4b Experimenteel ontwerp voor zeven factoren*)

*) In deze tabel zijn voor het gemak de enen weggelaten.

Ter toelichting nog het volgende: in de eerste drie kolommen staan de mogelijke combinaties voor x_1, x_2 en x_3 ($2^{7-4} = 2^3 = 8$). De kolom van x_4 is ontstaan door vermenigvuldiging van de tekens in de kolommen van x_1 en x_2. We schrijven dit aldus op: $x_4 = x_1x_2$. Evenzo is $x_5 = x_1x_3$, $x_6 = x_2x_3$ en $x_7 = x_1x_2x_3$.

We zien dus, dat door de hier gedemonstreerde kunstgreep het aantal experimenten krachtig kan worden beperkt. Zo'n beperkte proefopzet wordt ook wel aangeduid als een *fractioneel experimenteel ontwerp* (Engels: fractional factorial design). Als alle mogelijke experimenten worden uitgevoerd, wordt gesproken van een *volledig experimenteel ontwerp* (Engels: full factorial design).

Een fractioneel ontwerp kan tot nadeel hebben dat de effecten van de afzonderlijke variabelen onvoldoende worden herkend. Ze kunnen als het ware onder één noemer worden samengebracht. Door een nauwgezette analyse van de factoren kan de onderzoeker een zodanig fractioneel ontwerp opzetten, dat de belangrijkste effecten worden herkend.

In bepaalde gevallen is het mogelijk, dat de factoren elkaar onderling beïnvloeden. Dit betekent dat in de responsevariabele naast de bijdrage van de afzonderlijke factoren ook het effect van de wederzijdse invloed, het zogenaamde *interactie-effect* is verwerkt. De regressievergelijking is dan niet van de eerste orde, zoals vergelijking IV.1, maar van een hogere orde. Deze laatste kan in het geval van twee factoren als volgt in een regressievergelijking worden weergegeven:

$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2 \qquad \text{(vergelijking IV.2)}$$

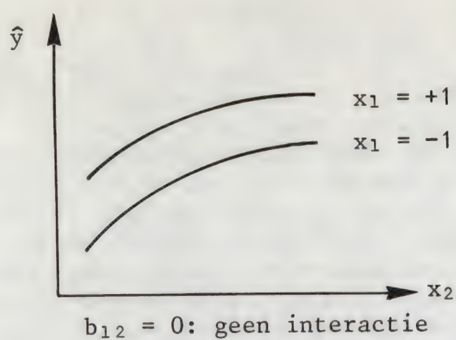
Een grafische voorstelling van interacties in zo'n situatie van twee factoren is te vinden in figuur IV.4c.

In deze paragraaf is de strategische planning van het experimenteel ontwerp behandeld. Hierbij is onderscheid gemaakt tussen het volledige (of structurele) en het functionele experimenteel model. Bij dit onderscheid gaat het om de factoren die van het meeste belang worden gedacht voor de verklaring van het systeemgedrag. Beperking van het aantal factoren zal leiden tot beperking van het aantal experimenten.

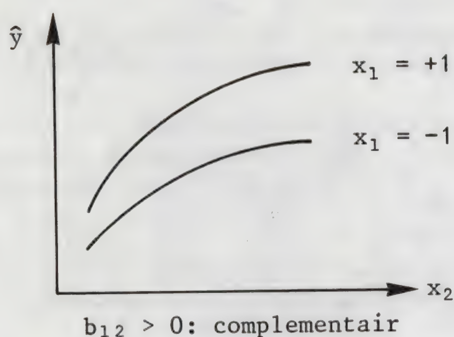
Naast het onderscheid volledig/functioneel is het onderscheid 'fractional factorial design' en 'full factorial design' ingevoerd. Dit onderscheid moet worden gezien als een onderdeel van het functionele experimenteel model. Het heeft eveneens als doel het aantal experimenten te reduceren en toch voldoende informatie te verkrijgen in ingewikkelde situaties. Hiertoe wordt gebruik gemaakt van zogenaamde statistische proefopzetten. Bij zo'n statistische proefopzet worden niet alle combinaties van factoren ('full factorial') onderzocht, maar slechts een deel ('fractional factorial'). Deze methode levert een grote accuratesse van de schatters van de factoreffecten*) en vereist minder experimenten.

---

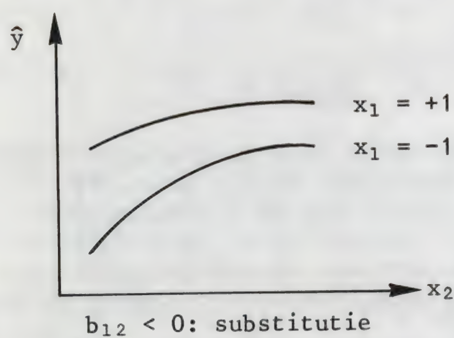
*) Zie: diverse publicaties van J.P.C. Kleijnen, vakgroep Informatica, Fac. der Economische Wetenschappen, Katholieke Hogeschool Tilburg.



$b_{12} = 0$: geen interactie



$b_{12} > 0$: complementair



$b_{12} < 0$: substitutie

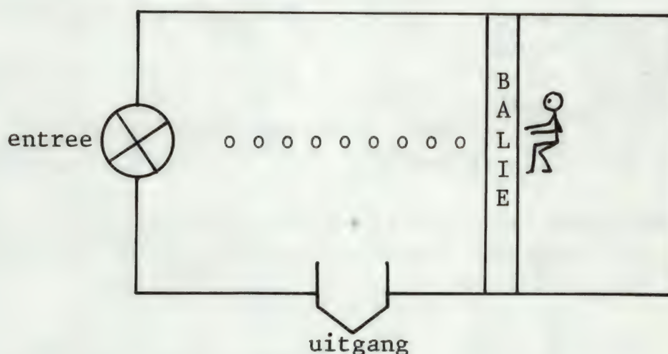
Figuur IV.4c Interacties

IV.5 DE TACTISCHE PLANNING

In § IV.1 hebben wij het experimenteel ontwerp onderscheiden in de strategische en de tactische planning. Waar het bij de strategische planning gaat om de vaststelling van de vereiste experimenten ten behoeve van de benodigde informatie, *gaat het bij de tactische planning om de efficiënte uitvoering van de experimenten.* Hoe lang moet, bijvoorbeeld, een bepaald experiment op de computer worden nagebootst? Computertijd heeft een prijs en kan niet onbeperkt worden gebruikt. De tijdsduur van een experiment op de computer hangt onder andere samen met de beginwaarden (≡ startwaarden) van de toestandsvariabelen. Zo zal de beddenbezetting in een ziekenhuis vrij zeker nooit nul zijn. Het is daarom in het algemeen niet efficiënt een ziekenhuissimulatie 'leeg' te starten. Alvorens dan namelijk een normale bezetting tijdens de computersimulatie is bereikt, is er al heel wat computertijd voorbij.

In feite is de tactische planning de fase waarin de 'technische parameters' van het experimentele model worden ingevuld. Anders gezegd: na de vraag van het 'wat' van de strategische planning is nu de vraag van het 'hoe' aan de orde. Aan de hand van een eenvoudig voorbeeld zullen we een aantal vraagstukken van de tactische planning aan de orde stellen.

We kiezen een eenvoudige loketsituatie*) met slechts één wachtrij, zoals in een klein postkantoor zou kunnen voorkomen.



De bedieningstijden en aankomsttijden zijn gegeven in de volgende tabel:

*) Zie K. Bauknecht e.a., Simulationstechnik, Berlin, 1976.

bedieningstijd in minuten	frequentie in %	verdelingsfunctie
0-1	30	0.3
1-2	40	0.7
2-3	20	0.9
3	10	1.0
tussen-aankomsttijd		
0.2	50	0.5
2-4	20	0.7
4-6	15	0.85
6-8	10	0.95
8-10	5	1.00

Tabel IV.6a*)

Welke responsevariabelen kunnen voor een onderzoek in aanmerking komen?

Afhankelijk van de doelstellingen kunnen verschillende mogelijkheden of combinaties ervan optreden:

- de wachttijd van de klanten tot aan het moment waarop de bediening aanvangt
- de wachttijd van de klanten tot zij het systeem verlaten, dus inclusief de helptijd
- de lengte van de wachtrij en de spreiding hierin
- de tijd dat het loket vrij is.

Welke problemen zal men eerst moeten oplossen, voordat men het model door de computer kan laten verwerken? We noemen een aantal.

1. *De waarnemingsperiode*
   Het lijkt logisch als waarnemingsperiode een werkdag van acht uur te kiezen. Toch zou een reële situatie kunnen zijn, dat er een middagpauze van bijvoorbeeld 12.30 uur tot 14.00 uur bestaat. Hiermee samenhangend is het mogelijk dat er 's middags een ander aankomstpatroon en/of een ander type klanten bestaat.

2. *De 'run' lengte*
Simuleert men dit systeem voor acht dagen, één maand of één jaar? Men dient bij deze vraag overwegingen in beschouwing te nemen, zoals seizoensinvloeden, periode binnen een maand (bijvoorbeeld aan het begin van de maand is er een groter klantenaanbod), de dag van de week (vrijdag is altijd de drukste dag). Wanneer men het aantal dagen als herhalingsmaatstaf neemt, dan wel als steekproefomvang, zal men met statistische technieken dit aantal moeten bepalen.

3. *De steekproefomvang*
Bij de bespreking van de 'run'lengte is al gebleken, dat de problematiek van de steekproefgrootte en runlengte kan samenvallen, afhankelijk van het systeem. Bij het loketsysteem zou men ook een vast aantal klanten, bijvoorbeeld 500 kunnen nemen als steekproefgrootte. De vaststelling van dat aantal geschiedt eveneens met statistische ondersteuning.

4. *De startwaarden*
Hieronder verstaat men het initialiseren van de verschillende toestandsvariabelen. Het ligt voor de hand om bij een loketsysteem de dagelijkse 'run' met een 'leeg' systeem te starten. De toestandsvariabelen (rijlengte, totale wachttijd, leeglooptijd van de loketbediende, enzovoorts) worden op 0 gezet. Niet bij alle systemen ligt dit probleem even eenvoudig (§ IV.7).

5. *De aanloopperiode (Engels: transient period)*
Waarschijnlijk zijn we in het gedrag van ons postkantoor geïnteresseerd in een situatie, waarbij er van een zekere stabiliteit (zo die ooit bereikt wordt) sprake is. De aanloopperiode verstoort het beeld. De wachttijden direct na de opening zullen immers nog gering zijn. M.a.w. we dienen het vraagstuk op te lossen, wanneer een zekere evenwichtssituatie bereikt wordt. Een niet onbelangrijke vraag is: wat is de evenwichtssituatie? (Zie § IV.7.)

6. *Afhankelijkheid*
De waarnemingen die we in het postkantoor doen, bijvoorbeeld in de vorm van wachttijden, zijn niet onderling onafhankelijk. Optredende correlatie en autocorrelatie zijn verschijnselen, die onderkend moeten worden en zullen moeten worden opgelost met behulp van de geëigende statistische technieken. Dit geldt ook voor verschillende categorieën uitkomsten. Neem bij het loket de volgende groepen waarnemingen:
- wachttijd klant
- de tijd, dat de klant in het systeem aanwezig is
- de lengte van de wachtrij.

Deze groepen zijn ten opzichte van elkaar evenmin statistisch onafhankelijk. Hieruit volgt, dat afzonderlijke analyse van

een van de drie responses 'gevaarlijk' is voor het trekken van juiste en betrouwbare conclusies.

7. *De toevalsgeneratoren*
Het vraagstuk van het trekken van toevalscijfers voor de stochastische variabelen (aankomsttijd, helptijd) valt uiteen in minstens drie deelvragen:
- kan men beschikken over faciliteiten voor het trekken van toevalscijfers uit een aantal verdelingsfuncties?
- zijn de generatoren getest en betrouwbaar?
- kan men (bijvoorbeeld met startwaarden voor de toevalsgeneratoren) meerdere parallelle reeksen toevalscijfers verkrijgen? (Zie § IV.6.)

8. *Stopregels*
Aan het einde van de simulatiedag moet men het systeem netjes afwerken. Dat wil zeggen: men laat geen nieuwe klanten meer toe. Alle zich 'binnen' bevindende klanten worden geholpen. Het is niet onmogelijk dat dit niet in overeenstemming is met de werkelijke gang van zaken. Simulatietechnisch kent men de stopregel, waarbij men programmatisch het einde van de run bepaalt, namelijk als de informatie aan bepaalde vastgestelde betrouwbaarheidscriteria voldoet.

Het zal na de opsomming van deze acht punten duidelijk zijn, dat er *geen eenduidige regels* te geven zijn voor het opstellen van een tactisch plan. Daarvoor is de problematiek te zeer afhankelijk van de aard en de doelstellingen van het onderzoek, de wijze waarop men de experimenten wenst op te zetten en van de validatie.

Bijna alle vraagstukken van de tactische planning worden veroorzaakt door de variabiliteit, door het feit dat de meeste simulatiemodellen stochastisch van aard zijn. Herhalingen in het experiment zijn dus noodzakelijk om betrouwbare uitkomsten te krijgen. Doch we dienen uit een economisch oogpunt (computertijd en mankracht zijn niet gratis) een minimale hoeveelheid 'runs' te bepalen om de grootst mogelijke hoeveelheid informatie te krijgen, welke voldoet aan de gestelde eisen ten aanzien van de betrouwbaarheid.

Dat betekent nadrukkelijk dat de onderzoeker zijn experimenten zo moet inrichten dat hij zinvolle uitspraken (m.b.v. statistische technieken) kan doen omtrent de betrouwbaarheidsintervallen, waarin de waarden van de responsevariabelen liggen.

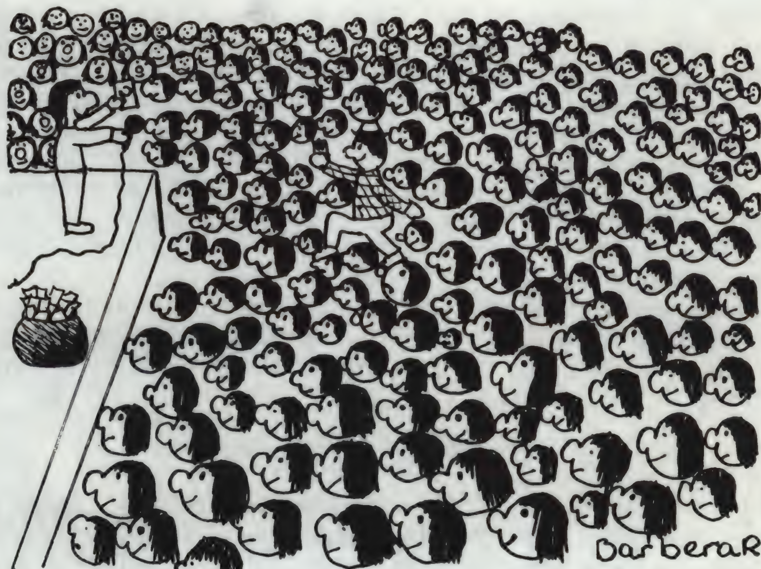
Tevoren moet vaststaan welk risico men wil aanvaarden om een bepaalde hypothese te verwerpen, terwijl deze juist is (in de statistiek aangeduid als *risico van de eerste soort* of fout van het type α) en het risico om een hypothese die onjuist is, niet te verwerpen (in de statistiek aangeduid als *risico van de tweede soort* of fout van het type β).

De nauwkeurigheid en de simulatie-inspanningen die men zich moet getroosten zijn nauw aan elkaar gerelateerd. Kan men door het probleem zorgvuldig te bekijken de verhouding hiertussen verbeteren, dus zonder meer runs de precisie verhogen, dan is dat kostenreducerend. Variantiereductietechnieken staan ons hierbij ten dienste om het gewenste effect te bereiken.

Een verhoging van de nauwkeurigheid kan ook bereikt worden door het aantal herhalingen per experiment te vergroten. Helaas is dit kostbaar, want willen we de fout van het gemiddelde terugbrengen tot de helft, dan moet het aantal waarnemingspunten verviervoudigd worden. De betrouwbaarheid is namelijk sterk afhankelijk van de tijdens de tactische planning gekozen waarden. Het aantal malen, dat eenzelfde experiment wordt uitgevoerd, de waarnemingsperiode, de steekproefomvang, zijn bepalende factoren voor de uitspraken, die men aangaande de betrouwbaarheid wenst te doen.

De statistiek speelt in deze fase van de simulatie een belangrijke rol. Zoals in § IV.1 reeds is gezegd, is in dit boek geen plaats voor een uitgebreide behandeling van statistische technieken. In Appendix A geven wij een overzicht van een aantal technieken die bij de uitvoering van simulatie-experimenten van belang kunnen zijn. Voor wie dit niet voldoende is, verwijzen we naar een statistisch leerboek.

In de volgende paragrafen zal een aantal vraagstukken van de tactische planning nader worden behandeld.



---rijdiscipline: FIFO---

IV.6 DE ORGANISATIE VAN SIMULATIE-EXPERIMENTEN

Zoals op een aantal plaatsen in voorgaande hoofdstukken reeds naar voren is gekomen worden de steekproeven die we trekken met behulp van een computerprogramma 'automatisch' gegenereerd. Hiertoe moeten we wel de beschikking hebben over toevalsgeneratoren. Door manipulatie met deze generatoren is het mogelijk verschillende parallelle reeksen waarnemingen te verkrijgen.

Bij een simulatieprogramma moeten drie elementen bepaald worden, waarmee het experiment geheel is vastgelegd:

1. de modelparameters (P)
2. de starttoestand (S_0)
3. de startwaarden van de randomgeneratoren (r_0)

In de modelbouwfase en de strategische planningsfase zijn over de modelparameters P beslissingen genomen. Het functionele experimenteel model is vastgesteld op grond waarvan de onderzoekers denken de gewenste informatie te kunnen verkrijgen.

Voor de uitvoering van een simulatie-experiment vormt de zogenaamde starttoestand (S_0) een noodzakelijk gegeven. In een voorraadmodel bijvoorbeeld zal men van een zekere beginvoorraad uitgaan. Deze beginvoorraad behoeft niet nul te zijn. In veel situaties is een van nul verschillende beginvoorraad zelfs efficiënter.

De rijen toevalsgetallen die bij een experiment worden gebruikt, kunnen met behulp van startwaarden van de randomgeneratoren worden gestuurd. De startwaarde r_0 ('het zaadje' van de toevalsgenerator) bepaalt de volgende waarde van de toevalsgetallen $r_1, r_2, \ldots, r_e$. De startwaarde bepaalt als het ware de plaats in de reeks van toevalscijfers waar wordt begonnen. Een andere startwaarde levert een andere reeks van toevalscijfers; dezelfde startwaarde geeft steeds opnieuw dezelfde toevalscijfers. Dit laatste is prettig als men met een identieke omgeving verschillende systeemsituaties wil bestuderen.

Concluderend kan worden vastgesteld, dat de waarde van een responsevariabele Y wordt vastgelegd door de drie elementen P, S_0 en r_0 alsmede door de tijd. Een responsevariabele is dus een functie van deze elementen:

$$Y = Y(P, S_0, r_0, t)$$

Als gevolg van de mogelijkheid het simulatieproces aan de hand van de vier genoemde elementen te bestuderen, kunnen experimenten op verschillende wijze worden uitgevoerd. Zo is het mogelijk bij dezelfde P en S_0 verschillende startwaarden voor de randomgenerator te kiezen. In dit geval spreken we van *herhalingen* (Engels: replications) van runs.

Bij het toepassen van herhalingen in een simulatie-onderzoek hoeven de herhalingen niet statistisch onafhankelijk te zijn. Dit is alleen het geval als de toevalsgetalrijen elkaar niet overlappen. Men kan de onafhankelijkheid op een eenvoudige wijze verkrijgen door de rij toevalsgetallen door te laten lopen. Voor de startwaarde van de herhaling wordt dan de eindwaarde van de voorgaande run gebruikt. De toevalsgetallenrij in de herhaling is dan een voortzetting van de toevalsrij van de voorgaande run. Een essentiële voorwaarde voor het slagen van dit middel is natuurlijk wel, dat de periodelengte van de toegepaste randomgenerator voldoende lang is, zodat geen herhaling optreedt van reeds gebruikte toevalscijfers.

Wanneer een eerste run stopt in een eindtoestand S_e^1 en het laatste toevalsgetal is r_e, dan moet een tweede run worden uitgevoerd met dezelfde modelparameters, doch met een begintoestand S_0^2 gelijk aan de vorige eindtoestand S_e^1 en de startwaarde van de randomgenerator r_0, eveneens gelijk aan de vorige eindwaarde van de rij, r_e. Een dergelijke simulatierun wordt een *voortzetting* genoemd.

Veronderstel dat in het voorbeeld van het loket in de vorige paragraaf een eind wordt bereikt, nadat 20 klanten geholpen zijn en dat op grond van de gegenereerde resultaten uitspraken worden gedaan. Men kan zich voorstellen, dat het bezwaar geopperd wordt dat de simulatie-'run' te klein is en de resultaten niet betrouwbaar genoeg zijn om de conclusies te accepteren. Dan kan men de simulatie weer helemaal opnieuw starten en het afbraakpunt een eind verder in de tijd leggen. Men kan echter ook doorgaan op het punt waar was afgebroken en aldus een voortzetting bewerkstelligen.

Let wel: deze mogelijkheid moet in het programma ingebouwd zijn, doordat men bereikte eindtoestanden zodanig opslaat, dat ze voor eventuele voortzettingen weer als starttoestanden in aanmerking kunnen komen.

Overigens heeft voortzetting niet altijd zin; voor het geval er een 'natuurlijke' eindtoestand is ontstaan, welke zich niet meer (significant) wijzigt, zal verdere simulatie zinloos zijn.

Tenslotte kan men wijzigingen aanbrengen in de verzameling modelparameters P; we spreken dan van *vergelijkingsruns*. Deze zullen evenmin per se statistisch onafhankelijk zijn, zoals bij de herhalingen. Wenst men dat wel, dan is het in ieder geval aan te raden doorlopende toevalsgetalrijen te genereren. Men zou zich echter ook kunnen uitspreken voor een sterke samenhang van de vergelijkingsruns. Men gebruikt dan precies dezelfde toevalsgetallen. Neem aan dat we in het voorbeeld van het loket het systeem willen uitrusten met een tweede loket. Dan zullen we, om zinvolle uitspraken ter vergelijking te kunnen doen, het liefst hetzelfde klantenaankomstpatroon zien. In dat geval dienen we een aparte randomgenerator voor de aankomsten en voor de bedieningstijden in te richten.

IV.7 STARTCONDITIES EN DE EVENWICHTSSITUATIE

Eén van de opmerkelijke verschillen tussen computersimulatie en simulatie in een laboratorium of in een 'real life' omgeving is het gemak, waarmee een experiment met een computer onder dezelfde omstandigheden kan worden herhaald of tijdens de uitvoering kan worden gestopt en kan worden hervat. De onderzoeker beschikt dus, zoals in de vorige paragraaf uiteengezet is over een belangrijk middel om experimenten te sturen. Dit betekent overigens niet, dat hij nu maar lukraak aan de computersimulatie moet beginnen. Integendeel, hij moet tevoren goed nadenken over de startcondities en over de vraag op welk moment hij informatie aan het experiment gaat ontlenen.

Gewoonlijk zijn we als onderzoekers geïnteresseerd in de verrichtingen van een systeem in de evenwichtstoestand. Alvorens het systeem deze toestand heeft bereikt, moet het eerst 'warm lopen'. De eerste fase van de simulatie geeft vaak geen getrouw beeld van de werkelijke gang van zaken. Het duurt even alvorens een systeem de normale toestand heeft bereikt.

Vaak wordt gedacht, dat een computersimulatie moet beginnen met een leeg systeem, dat wil zeggen op het moment dat er nog geen acties gaande zijn. De eerste handelingen moeten dan nog een aanvang nemen. Of dit echter een goed uitgangspunt is, hangt helemaal af van de werkelijkheid die we willen nabootsen en van de doelstellingen van de simulatie. Simulaties van een ziekenhuis, een luchthaven of een goederenmagazijn zullen niet altijd 'leeg' dienen te starten in tegenstelling tot simulaties van een loketsysteem in een postkantoor of in een bank.

De periode die voorafgaat aan de relevante periode van de simulatie, duiden we aan als voorloop of inloopperiode. Om het verstorende effect van deze periode op de resultaten van de simulatie te verminderen kunnen verschillende adviezen worden gegeven.

1. Gebruik computerruns die lang genoeg zijn, zodat de gegevens uit de beginperiode geen significante invloed hebben op de resultaten van de computersimulatie in de evenwichtssituatie.

Deze methode is redelijk bevredigend als een computerrun weinig tijd vraagt, zodat een grote steekproefomvang geen bezwaar is. Als het model echter complex is en iedere uitkomst kostbaar is te verkrijgen of als de inloopperiode relatief veel tijd in beslag neemt, dan wordt deze methode twijfelachtig.

2. Start de computersimulatie, maar laat voor de berekening van de vereiste informatie de gegevens uit de voorloopperiode buiten beschouwing. Ook deze methode brengt wel een aantal problemen met zich.

In de eerste plaats gebruiken we een gedeelte van de gegevens dus niet voor de berekeningen. Toch is met dit doel van de computerrun (vaak kostbare) computertijd gemoeid. Ten tweede kan deze methode de variantie van de resultaten verhogen, hetgeen de nauw-

keurigheid van de resultaten ongunstig beïnvloedt. Ten derde blijft de hamvraag: wat is de lengte van de voorloopperiode? Er bestaat namelijk geen waterdichte methode om het bereiken van de evenwichtssituatie (Engels: steady state) te signaleren, temeer daar deze situatie niet in een constante waarde tot uitdrukking wordt gebracht. De waarde(n) van de responsevariabele(n) zullen altijd schommelen rond gemiddelden.

We nemen aan dat er voor ieder model een waarschijnlijkheidsverdeling van de responsevariabele bestaat, die karakteristiek is voor het systeem. Een model heeft de evenwichtssituatie bereikt als de uitkomsten overeenkomen met die verdeling. Wanneer we, zoals ook wel voorkomt, niet in de evenwichtssituatie geïnteresseerd zijn, maar juist in de verrichtingen van het systeem die naar het evenwicht toe leiden, is het probleem identiek.

De methoden, die ter oplossing aangewend worden en meer het karakter van vuistregels hebben, zijn gewoonlijk gebaseerd op het beschouwen van een 'proefrun' (Engels: pilot run). Zo mogelijk dient men zijn beslissingen op grond van de resultaten van meer 'proefruns' te nemen. De instabiliteitsperiode is namelijk afhankelijk van de startwaarde van de randomgeneratoren.

We noemen een aantal suggesties, die door verschillende onderzoekers naar voren zijn gebracht.

a. Breek een rij metingen af, als de eerste meting van de reeks noch het maximum noch het minimum van de overblijvende verzameling is.

b. Als het aantal waarnemingen, waarvan de uitkomsten groter zijn dan het gemiddelde, ongeveer even groot is als het aantal uitkomsten kleiner dan het gemiddelde, dan is het waarschijnlijk dat 'steady-state' condities zijn bereikt.

c. Bereken het voortschrijdend gemiddelde van de uitkomsten en neem aan dat het evenwicht bereikt is, wanneer het gemiddelde niet meer significant verandert.

Met name de laatste suggestie biedt een aanvaardbaar hulpmiddel voor de beoordeling van de evenwichtssituatie. De beoordeling van de mate van significantie is evenwel subjectief en is mede afhankelijk van de doelstellingen van het onderzoek. De twee eerstgenoemde suggesties zullen in het algemeen te weinig houvast bieden.

3. Kies startcondities die min of meer karakteristiek zijn voor de evenwichtssituatie. Hierdoor kan het voorloopstuk wel bekort worden, doch niet geheel geëlimineerd. Het gevaar dat in deze methode schuilt is, dat we gegevens krijgen die we zelf beïnvloed hebben door de keuze van de startwaarden. Het is altijd aan te raden de invloed van de startwaarden op de responsevariabelen expliciet te maken.

Vaak wordt simulatie toegepast voor vergelijking van twee of meer alternatieve gedragslijnen in beslissingssituaties. Stel, dat we twee alternatieven (bijvoorbeeld één of twee loketten in een post-

kantoor) willen vergelijken met elk geheel verschillende evenwichtscondities. Welke startcondities kunnen nu het beste worden gekozen?

De onderzoeker beschikt voor de beantwoording van deze vraag over drie mogelijkheden:

- elk van de systemen simuleren met een 'lege' uitgangssituatie;
- elk van de systemen simuleren met startwaarden die een compromis zijn tussen de meest acceptabele startwaarden van elk systeem afzonderlijk;
- elk van de systemen simuleren met de eigen meest voor de hand liggende startwaarden.

Uit efficiency-overwegingen heeft de tweede mogelijkheid de voorkeur boven de eerste en de derde boven de tweede. We moeten evenwel oppassen nu we met de tactische planning bezig zijn onze aandacht alleen te richten op het efficiencyvraagstuk. De derde mogelijkheid heeft namelijk als mogelijk nadeel, dat de onderzoeker bij voorbaat de resultaten van het onderzoek beïnvloedt. Het lijkt immers logisch, dat de simulatie een bepaald resultaat oplevert als reeds bij de aanvang door de keuze van de startwaarden hierop geanticipeerd is. Vandaar, dat de tweede mogelijkheid in het algemeen als de meest aanvaardbare moet worden aangemerkt.

Als slotconclusie van deze paragraaf willen we vaststellen, dat het efficiencyvraagstuk bij simulatie van wezenlijk belang is, maar dat dit vraagstuk nooit ten koste mag gaan van de validiteit van het onderzoek. Het is niet zo moeilijk om onder het mom van wetenschappelijk onderzoek indrukwekkende resultaten te presenteren. De validiteit van de resultaten mag evenwel nooit in het geding komen door de keuze van bepaalde startwaarden.

IV.8 DE RUNLENGTE; AUTOCORRELATIE

In de simulatiepraktijk wordt het aantal waarnemingen van een run vaak vastgesteld op een mooi getal, een veelvoud van tien of van honderd. De overwegingen die tot dit als runlengte aangeduid getal leiden, worden veelal niet expliciet gemaakt. In hoeverre de gemaakte keuze in verband met de vereiste betrouwbaarheid en nauwkeurigheid juist is, zal dan afhangen van het toeval. Het aantal waarnemingen zal dan veelal of te groot of te klein zijn. Terwijl in het geval van een te grote runlengte er sprake zal zijn van inefficiency, zal in het geval van een te kleine runlengte de run vaak geheel 'verloren' blijken te zijn.

In de praktijk zijn simulatiemodellen doorgaans begrensd, dat wil zeggen dat de run wordt beëindigd als een specifieke situatie optreedt. Aan Kleynen*) ontlenen wij een aantal voorbeelden van specifieke situaties die tot beëindiging van een simulatierun leiden.

1. Bij het onderzoek naar het opstellen van een geschikt onderhoudschema van machines of andere apparatuur (bijvoorbeeld een computer) eindigt de simulatierun wanneer het betreffende apparaat uitvalt door een defect. Een nieuwe run begint bij een serie goed functionerende apparaten.

2. Een systeem met wachtrijen, zoals een postkantoor of de haringkar heeft een sluitingstijd om bijvoorbeeld 6 uur 's middags. De nieuwe run begint op de volgende dag, en start dan 'leeg'.

3. Een simulatie van een ondernemingsmodel (Engels: corporate model) kan gehanteerd worden om bepaalde besturingsrichtlijnen te vergelijken. Bijvoorbeeld: het effect op de omzet over de komende 3 maanden (de planningshorizon) wordt bestudeerd door de simulatie te herhalen voor verschillende richtlijnen. Elke run wordt gestart vanaf de meest recente reële situatie in het systeem. (Vaak zijn ondernemingsmodellen niet stochastisch!)

4. Er zijn ook wachtrijsystemen die 'nooit' ophouden op een natuurlijke wijze, zoals een verkeerskruispunt op drukke wegen of telefoonverkeer. Zulke systemen worden gesimuleerd om te onderzoeken of een systeem piekbelasting kan verdragen. Als de piektijd ophoudt, dan stopt ook de simulatierun. Een nieuwe run start vanaf de situatie die gold vóór de piektijd. In die volgende run kunnen zowel dezelfde of andere toevalsgetallen worden gebruikt, om zo de gevoeligheid voor de startcondities te onderzoeken (inclusief de startwaarde van de generator van de toevalsgetallen).

---

*) J.P.C. Kleijnen, The role of statistical methodology, in: Methodology in systems modeling and simulation, Amsterdam 1979.

In elk van deze voorbeelden maken de specifieke begin- en eindtoestand van het systeem deel uit van de simulatie en zijn dus in de responsevariabele verweven. Door de 'run' te herhalen met andere toevalscijfers kunnen nauwkeuriger schattingen omtrent de responsevariabele, bijvoorbeeld de wachttijd, worden verkregen.

Statistische berekeningen zijn nodig om de grenzen van de betrouwbaarheidsintervallen te bepalen. Raadpleeg de appendix om concrete formules te kunnen vinden. De nauwkeurigheid van de responsevariabelen kan worden vergroot, zodat het betrouwbaarheidsinterval wordt verkleind, door de runlengte te vergroten.

In de vorige paragraaf is de evenwichtssituatie van een simulatie-experiment ter sprake gekomen. We hebben toen betoogd, dat de simulatie eerst 'op gang' moet komen alvorens van een evenwichtssituatie kan worden gesproken. Dit 'op gang komen' kan tot consequentie hebben dat de run erg lang wordt, hetgeen de kans op afhankelijkheid tussen de afzonderlijke waarnemingen verhoogt. Immers, als in een postkantoor een klant erg lang moet wachten, dan moet de volgende klant waarschijnlijk ook erg lang wachten. Er is dan sprake van correlatie tussen de onderlinge wachttijden. We spreken in zo'n geval van *autoregressie* of ook wel van *autocorrelatie*.

Van autoregressie is sprake wanneer de omvang van de fluctuatie in de periode t deels kan worden verklaard uit de omvang in een voorgaande periode, bijvoorbeeld $t-1$ of $t-2$ en wel *van dezelfde variabele*. Men spreekt van autocorrelatie als er in een reeks *toevalstermen* tussen de toevalstermen onderling correlatie bestaat. Wij zullen in de rest van de tekst alleen de term autocorrelatie gebruiken, dus ook in de gevallen dat er eigenlijk sprake is van autoregressie.

Men kan autocorrelatie op het spoor komen door de afwijkingen van het gemiddelde van een reeks waarnemingsuitkomsten vast te stellen en na te gaan of er langere reeksen van positieve dan wel van negatieve afwijkingen voorkomen. Is dit het geval of wil men meer zekerheid, dan kan een autocorrelatiecoëfficiënt worden berekend (zie de statistische appendix).

Inzicht in de mate van autocorrelatie is belangrijk, omdat bij aanwezigheid hiervan de statistische technieken die uitgaan van onafhankelijkheid van de waarnemingsuitkomsten niet meer toegepast kunnen worden. De variantie van een response-variabele bijvoorbeeld wordt veel hoger bij afhankelijke uitkomsten dan bij onafhankelijke.

Er bestaan verschillende methoden om het autocorrelatieprobleem terug te dringen.

a. De methode van de herhaling (Engels: replicated or repeated runs)

Deze methode bestaat uit het uitvoeren van k onafhankelijk runs elk met m waarnemingen. De onafhankelijkheid ontstaat door bij elke run een nieuwe startwaarde voor de toevalsgenerator te nemen. De methode heeft als nadeel de voorloopfase bij elke run.

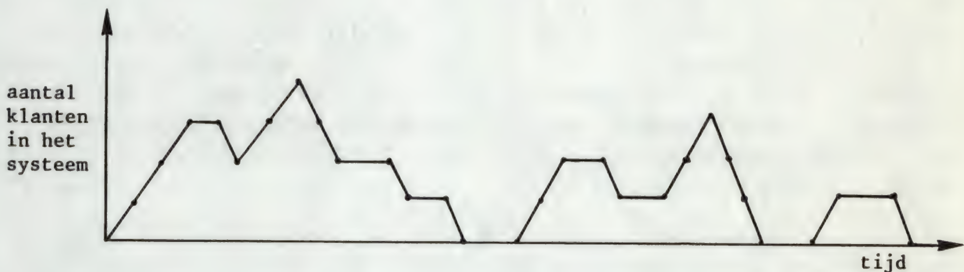
b. De methode van de voortgezette run

We maken één doorlopende lange run; de waarnemingen worden verdeeld in k 'subruns'. De eerste voorlooprun laten we buiten beschouwing. We beschouwen $\overline{Y}_i$ $(i = 1,2,\ldots,k)$ als de uitkomst van herhaling i. Als de lengte van de 'subrun' (m waarnemingen) groot genoeg is, dan zijn de $\overline{Y}_i$ als onafhankelijk te beschouwen. Bovendien zijn voor een behoorlijk grote m de $\overline{Y}_i$ bij benadering normaal verdeeld.

Deze methode verdient t.o.v. de vorige de voorkeur waar de initiële problemen grotere betekenis hebben. Deze methode is geschikt voor 'non-terminating' systemen. Convergeren de systemen wel naar een onveranderlijke eindtoestand (Engels: terminating systems), dan kan men beter de methode van de herhalingen gebruiken.

c. De regeneratieve methode

Deze (flexibele) methode is geschikt voor systemen, die een zekere regeneratie in zich hebben, waarbij op willekeurige tijden toestanden ontstaan die al eerder geweest zijn, bijvoorbeeld de starttoestand. In een loketsysteem, dat zeker leeg zal starten, zullen na verloop van tijd ongetwijfeld alle klanten geholpen zijn, terwijl er nog geen nieuwe klanten gearriveerd zijn. We krijgen dan het volgende beeld:



Tussen twee opeenvolgende regeneratie-tijdstippen zijn de uitvoervariabelen zeker onafhankelijk.

Hierbij geldt overigens wel als voorwaarde dat het aankomstproces 'geheugenloos' moet zijn, zoals bij een Poissonproces.

IV.9 NOGMAALS DE HARINGKAR

Zoals in § 4 van dit hoofdstuk is toegezegd komen we hier nog eens terug op het voorbeeld van de haringkar. Het in figuur IV.4b gepresenteerde functioneel experimenteel model resulteerde in vier experimenten.

Voor de uitvoering van de simulatie kiezen we voor de methode van de herhaling, daarmee aansluitend bij het reële proces, dat elke dag een beeld geeft te zien van acht uur haringverkoop. Elke herhaling is dus een dag. Elke herhaling levert informatie omtrent het bedieningsproces op een bepaalde dag. Zo kunnen we de gemiddelde wachttijd met de standaardafwijking per dag berekenen. We kunnen een histogram van de gemiddelde wachttijd over een aantal dagen maken. Andere interessante informatie is de maximale wachttijd, het aantal keren per dag dat de wachttijd boven een aanvaardbaar maximum komt en de gemiddelde rijlengte.

Door de keuze van de methode van de herhaling starten we iedere dag met een leeg systeem. Het probleem van de startwaarden ontlopen we door deze keus. Een belangrijke vraag in het kader van de tactische planning is nu: hoeveel dagen nemen we als steekproefgrootte?

Voor de berekening van de steekproefomvang moeten we allereerst de (hoofd-)uitvoervariabele vaststellen. In het geval van de haringkar kiezen we voor de gemiddelde wachttijd per dag. Bij uitvoering van het experiment krijgen we dan voor n dagen een gemiddelde (symbool $\overline{w}_i$ voor $i = 1,2,...,n$). Behalve de gemiddelde wachttijd is de standaardafwijking per dag van belang i.v.m. de spreiding van de wachttijden rond het gemiddelde. We krijgen dan n verschillende standaardafwijkingen, namelijk S_i voor $i = 1,2,...,n$.

In de statistische appendix wordt een aantal formules gegeven voor de berekening van de steekproefgrootte (het aantal dagen dat we de haringkar simuleren). In het geval dat de gemiddelde wachttijden $\overline{w}_i$ normaal verdeeld zijn bij onbekende standaardafwijking, is de formule:

$$n \geqq \frac{t^2 \cdot s^2}{d^2}$$

van toepassing. In deze formule moet $d = b \cdot \overline{W}$ worden ingevuld overeenkomstig de eisen van nauwkeurigheid op grond van de doelstelling(en) van het onderzoek. s is de standaardafwijking van de gemiddelde wachttijden per dag rond het generale gemiddelde $\overline{W}$.

Als $\overline{W}$ bijvoorbeeld 4 is en als we een fout van 5% willen toelaten, dan is:

$$d = b*\overline{W} = 0,05*4 = 0,2$$

Als verder $s \approx 0,7$ en $t \approx 2$, dan kan door invulling in de vergelijking worden gevonden:

$$n \geq \frac{4*0,49}{0,04} = 49$$

Aan deze benaderingsmethode kleven wel enkele bezwaren. In de eerste plaats is de waarde van de standaardafwijking tevoren niet bekend en in de tweede plaats is de waarde van t mede afhankelijk van het aantal waarnemingen.

Eén van de mogelijke oplossingen voor deze problematiek is het uitvoeren van een proefrun, van een testfase. Aan de hand hiervan kan enig inzicht worden verkregen in de waarde van de onbekende grootheden. Gelukkig is het zo, dat de invloed van een fout in de schatting van s of t op de steekproefomvang niet zo groot is. Van veel grotere invloed is de waarde van d. Juist de keuze door de onderzoeker van het nauwkeurigheidspercentage is voor de grootte van n belangrijk.

In § 4 is in het functioneel experimenteel model sprake van vier experimenten. Enerzijds willen we voor normale dagen de invloed van het aantal (één of twee) bedienden op de gemiddelde wachttijd vergelijken en anderzijds hetzelfde voor de vrijdag. Voor het vergelijken van de normale dagen ligt het voor de hand voor beide experimenten hetzelfde aantal dagen te simuleren. Dat geldt eveneens voor de beide experimenten van de vrijdag. Overigens behoeft het aantal simulatiedagen van de ene groep experimenten niet hetzelfde te zijn als voor de tweede groep van de vrijdag. (Verklaar dit.)

Voor elk van de dagen van de simulatie geldt dat klanten die aan het eind van de dag nog in de rij staan, worden geholpen. Er worden dan geen nieuwe klanten aan de wachtrij toegevoegd. Elke nieuwe dag zal met een andere rij toevalsgetallen worden gesimuleerd; deze rij kan een doorlopende reeks zijn uit de toevalsgenerator. De twee experimenten per groep moeten in verband met de vergelijkbaarheid van de uitkomsten wel met dezelfde reeks toevalsgetallen worden uitgevoerd.

IV.10 OPGAVEN

Opgave 1

Stel, men doet een onderzoek in vier scholen voor HEAO, die vier afdelinen kennen (BE, BI, CE en EJ).
De studenten worden onderscheiden naar geslacht (m, v) en naar inkomensgroep van de ouders (laag, midden en hoger inkomen). Het onderzoek dient om uit te maken welke van de genoemde factoren van (meer of minder) invloed zijn op de kansen van de studenten om na hun afstuderen 'hogere posities' in het bedrijfsleven te kunnen vervullen.
Stel een structureel-experimenteel-ontwerpmodel op.
Ontwerp een full-factorial-design; ga daarbij in eerste instantie uit van acht herhalingen.
Om het totaal aantal computerruns te verminderen gaat men onderzoeken of een reductie van het functioneel model mogelijk is. Stel hiervoor criteria op om tot de gevraagde reductie te komen. Wat is uw uitgangspunt t.a.v. het aantal factoren en hun waarden?
Pas een en ander toe op het bovengenoemd probleem.
Geef kort een beschouwing over de wenselijkheid om de verkregen reductie toe te passen.

*Opgave 2 *)*

Een vloot van 80 vissersschepen is normaal uitgerust voor de haringvangst. Door overheidsmaatregelen beknot in de mogelijkheden, overweegt de eigenaar van de vloot om op kabeljauw te gaan vissen. Daarnaast zou hij kunnen beslissen om op een andere vangstmethode (andere netten, bijvoorbeeld grofmazig) over te gaan, teneinde de opbrengst aan kabeljauw te vergroten. Met behulp van simulatie wordt een en ander onderzocht.
Er is dus sprake van een 2^2 design, waarbij we te maken hebben met 2 factoren x_1 en x_2, die elk twee waarden kunnen aannemen.
Men wil m.b.v. variantie-analyse het effect van de factoren onderzoeken.

a. Geef in een $2 * 2$ matrix het 'full factorial design' voor bovenstaand model.
b. Geef in een algemene formule-vorm weer, hoe de response Y afhangt van de beide factoren x_1 en x_2, maar laat de eventuele interactie buiten beschouwing.
c. De in b. gevraagde formule bevat enige onbekende parameters. De uitkomsten van het experiment leveren schatters voor die parameters op. Geef in een tabelvorm op overzichtelijke wijze, hoe de verwachting en de variantie van de genoemde schatters worden berekend uit de responsevariabelen.
(Noem deze Y_{00}, Y_{01}, Y_{10}, Y_{11}.)
d. Welke nulhypothese definieert u en hoe denkt u die te toetsen?
e. Beschrijf nu kort en globaal de term interactie.
Zou bij het vissersvlootmodel interactie optreden?
Bij welke waarde(n) van de factoren x_1 en x_2 wordt de interactie gemeten?
f. Wanneer gebruikt men variantie-analyse en wanneer regressie-analyse? Kunt u één of meer kenmerkende verschillen noemen?

---

*) Voor een onderdeel van dit vraagstuk is enige kennis vereist van variantie-analyse met meer dan één factor.

Opgave 3

Haringkarren zijn niet meer, zoals vroeger, een vertrouwd straatbeeld. Mensen, die van de visverkoop hun broodwinning maken, hebben tegenwoordig een viswinkel.
De viswinkel, die wij op het oog hebben, is tamelijk uitgebreid met een groot assortiment. De toonbank is verdeeld in een drietal compartimenten: er is een ruimte voor de verkoop van verse vis, die op verzoek van de klant schoongemaakt of gefileerd wordt; vervolgens is er een soort snelbuffet vooral voor gebakken vis, die in speciale warmhoudvitrines wordt bewaard, doch bij verkoop geen aparte bewerking meer behoeft; tenslotte is er nog een 'hapjeshoek': broodjes met paling, groene haring in het goede seizoen (met uitjes) en garnalensnacks kunnen daar worden verkregen en indien gewenst meteen geconsumeerd.
De kassa staat apart: geen geld bij vis, is de stelregel van de eigenaar.

Onze viswinkel mag zich verheugen in een goede klandizie. Er is voor de eigenaar echter het probleem dat het klantenpatroon niet zo gelijkmatig over de dag verdeeld is, terwijl ook de dag in de week nogal bepalend is voor de trek in vis bij de klanten.

Het dagpatroon ziet er ongeveer als volgt uit:

9 uur - 10 uur	10 klanten	per uur gemiddeld
10 - 11	20 klanten	
11 - 13	30 klanten	
13 - 15	15 klanten	
15 - 16	20 klanten	
16 - 17	35 klanten	
17 - 18	25 klanten	

Dit geldt voor de maandag, dinsdag en woensdag.
Donderdag geeft $1\frac{1}{2}$ maal zoveel klanten, terwijl de topdag vrijdag 3 maal zoveel klanten in de winkel laat zien. Het dagpatroon is echter hetzelfde.
Op zaterdag komen er gemiddeld 2 maal zoveel klanten, doch nu is het uurpatroon afwijkend:
van 9 - 13 uur komen de mensen voor de 'gewone' boodschappen, namelijk verse vis en gebakken vis in de verhouding van 2 : 3, maar van 13 - 16 uur (om 16 uur sluit de zaak) is er veel liefhebberij voor de hapjeshoek: zeker 45 klanten per uur.
Verder is het dagelijkse patroon van verhoudingen over de drie compartimenten in de winkel gelijk:

tijd	verse vis	gebakken vis	hapjeshoek
9 - 12 uur	2	2	1
12 - 13 uur	1	2	4
13 - 16 uur	2	2	2
16 - 18 uur	1	3	5

De bedieningstijden verschillen ook nogal wat:
Voor verse vis ligt dat als volgt:

percentage	helptijd
10	0.5 - 2.5 minuut
30	2.5 - 5
30	5 - 8
20	8 - 12
10	12 - 20

Deze helptijden zijn nagenoeg uniform verdeeld.
Bij de gebakken vis-liefhebber ligt het wat eenvoudiger: de helptijd is Poisson verdeeld met een gemiddelde van 2 minuten en een bovengrens van 6 minuten. De hapjes klanten hebben genoeg aan een helptijd van 2 - 3 minuten (uniform).
Bij de kassa zijn de klanten ongeveer een minuut kwijt.

Het is duidelijk dat de eigenaar van de viswinkel het niet gemakkelijk vindt om een zodanig personeelsbestand te hebben, dat de klanten een goede service kunnen krijgen wat betreft hun wachttijden. Daarom vraagt hij met behulp van simulatie een overzicht te maken, waarmee hij per DAGDEEL (morgen en middag) kan zien hoeveel bedienden hij in de winkel moet hebben, bij wachttijden die in principe de 5 minuten niet overschrijden. Het is namelijk voor de eigenaar goed mogelijk om met parttimekrachten een variabel personeelsbestand te hebben.

Vragen en opdrachten.

1. Geef een specificatie van het systeem. Gebruik hierbij een schets. Maak de aannamen die u nodig vindt.

2. Formuleer de doelstellingen van de simulatie.
Specificeer de uitvoer van het simulatie-computermodel, waarmee aan de doelstelling kan worden voldaan.

3. Welke beslissingsvariabelen neemt u op in uw model?

4. Welke invoervariabelen neemt u in uw model op?
In welke vorm laat u het simulatiemodel de invoer genereren?
Met welke technieken bereikt u de gekozen vorm van de invoer?

5. Stel een volledig experimenteel ontwerp op.

6. Reduceer het experimenteel model totdat een functioneel hanteerbaar experimenteel model ontstaat.

7. Ontwerp een tactisch plan om de experimenten uit te voeren:
   - Wat is de eenheid van de steekproef?
   - Bepaal de steekproefgrootte.
   - Stel start- en stopregels op.
   - Zowel de herhaling als de voortgezette run moeten nu gebruikt worden. Hoe?
   - Los het 'aanloopprobleem' op.
   - Is er sprake van vergelijkingsruns en zo ja, wanneer?

HOOFDSTUK V

Van Model naar Programma

V.1 INLEIDING

Een simulatie-onderzoek zal in veel gevallen uitmonden in een simulatieprogramma. Het programmeren is één van de fasen in het totale simulatieproject. Alvorens te gaan programmeren is het noodzakelijk dat we beschikken over een behoorlijk simulatiemodel. Dat wil zeggen dat er duidelijkheid is over het probleem dat we willen oplossen. Het systeem is geanalyseerd en we weten welke informatie we nodig hebben om uitspraken te kunnen doen over het systeem. M.a.w. we beschikken over een zo volledig mogelijke systeemspecificatie, waarmee de programmeur aan het werk kan. Deze staat voor een zware taak. Hij moet in staat zijn om een - vaak ingewikkeld - programma te ontwikkelen, dat flexibel en goed gedocumenteerd is. In dit hoofdstuk beschouwen we eerst de hulpmiddelen waarover een programmeur moet beschikken.

De mogelijkheden op technisch terrein en de met de technische ontwikkelingen parallel lopende vooruitgang op het gebied van de programmeertalen hebben simulatie als onderzoekmethode enorm gestimuleerd. De uitspraak: "zonder computer geen simulatie" is wellicht iets te boud, zeker als simulatie in een bredere zin wordt ingevuld dan in dit boek gebeurt, maar dat de techniek als hulpmiddel een grote bijdrage levert aan de ontwikkelingen van simulatie is buiten kijf.

Allereerst besteden we enige aandacht aan de computer, een essentieel hulpmiddel voor de realisatie van een simulatietoepassing.

Naast de techniek is vooral de vooruitgang op het gebied van programmeertalen stimulerend voor de ontwikkeling van moderne onderzoekmethoden als simulatie. Na de machinecodes, de assembleertalen en de proceduregerichte (algemene) talen zijn er thans de zogenaamde probleemgerichte (bijzondere) talen. Een belangrijk voordeel van deze laatste groep talen is dat ze ontwikkeld zijn met als doel het programmeren in specifieke toepassingsgebieden te vereenvoudigen. Hierdoor kan o.a. de programmeringstijd in belangrijke mate worden beperkt.

In § 3 houden we ons bezig met algemene en specifieke kenmerken van programmeertalen en simulatietalen. We gaan in op eisen die aan talen kunnen worden gesteld en we belichten enige keuzecriteria.

In § 4 bespreken we een drietal benaderingswijzen voor de beschrijving van simulatiemodellen. De keuze van een simulatietaal hangt namelijk nauw samen met de benaderingswijze.

De programmeertalen BASIC en Pascal worden kort besproken in § 5. Deze talen zijn gekozen uit de grote groep van proceduregerichte talen omdat ze in het onderwijs veel worden toegepast. Ter illustratie is voor elk van de talen een computerprogramma voor de haringkar van hoofdstuk I toegevoegd.

In de §§ 6 en 7 wordt de simulatietaal SIMULA behandeld. Aan een aantal zeer bijzondere eigenschappen van deze taal wordt aandacht geschonken. Allereerst wordt in § 6 in algemene zin iets gezegd over de mogelijkheden van SIMULA, waarna deze mogelijkheden in § 7 aan de hand van een simulatieprogramma worden gedemonstreerd.

Tenslotte bespreken we in § 8 een aantal aanwijzingen om het simulatiemodel in een acceptabel computerprogramma om te zetten. Veel programmeurs begaan de fout om nog voordat het probleem goed is geanalyseerd al het programma te willen schrijven. Deze aanpak zal in het algemeen leiden tot ondeugdelijke programma's. Vandaar dat wij de in § 8 gegeven aanwijzingen met nadruk onder de aandacht van de lezers willen brengen.

V.2 DE COMPUTER

In het algemeen zal de onderzoeker niet zo bijster veel inspraak hebben in de keuze van de computerfaciliteiten; hij kan mogelijk zijn wensen naar voren brengen aangaande zaken als:

1. on-line faciliteiten, zodat hij direct met een geïmplementeerd model kan experimenteren;
2. de aanwezigheid van een of meer compilers van simulatietalen, bijvoorbeeld een SIMULA-compiler;
3. de aanwezigheid van een statistisch standaardpakket, bijvoorbeeld SPSS;
4. de geheugengrootte, opdat hij hiermee geen rekening hoeft te houden i.v.m. de afmetingen van de simulatiemodellen;
5. invoerings- en testfaciliteiten om zijn simulatieprogramma te kunnen ontwikkelen.

In veel organisaties is er tegenwoordig wel een computer beschikbaar. De onderzoeker zal zich dan op de hoogte moeten stellen van de mogelijkheden, beperkingen en knelpunten en hij zal daarmee rekening moeten houden bij de planning van het project (zie hoofdstuk VII).

Geheel afhankelijk van de omgeving waarin gewerkt wordt, bijvoorbeeld research laboratoria, universiteiten, banken, levensverzekeringsmaatschappijen, overheidsinstellingen of industrieel-technische ondernemingen (meet- en regeltechniek, elektronika), zal men probleemgebieden aantreffen, waarin ofwel discrete ofwel continue simulatie toegepast wordt. In relatie hiermee kan men in computertypen het verschil aantreffen tussen discrete en analoge (analoog-hybride) computers. Deze computertypen zien er geheel verschillend uit en worden ook geheel verschillend 'geprogrammeerd'.

Analoge computers zijn heel geschikt voor simulaties van mechanische, elektronische of chemische processen. In een dergelijke computer combineert men schakelingen waarmee men door het instellen van spanningen en stroomsterktes dezelfde wiskundige vergelijkingen concretiseert, die aan de te simuleren processen ten grondslag liggen. Vandaar de benaming 'analoog'.

Voor de ontwikkeling van een simulatiemodel maakt het principieel niet veel uit welk type model (continu of discreet) met behulp van welk type computer (analoog of discreet) wordt ontwikkeld: alle fasen dienen toch doorlopen te worden. Slechts het beschrijvingsprofiel van het model zal verschillen: de continue simulatiemodellen zijn dikwijls een verzameling differentiaalvergelijkingen, de discrete modellen worden in een procesbeschrijving (zie ook § V.4) weergegeven met vaak 'impliciete relaties'. Ook het programmeren is onvergelijkbaar: bij analoog continue computers programmeert men door het instellen van spanningsbronnen en meet-

en regelknoppen. We zullen verder geen aandacht besteden aan analoge computers, niet omdat ze niet zeer goed toepasbaar zouden zijn, maar vanwege het specialistische karakter.

De meer bekende vorm van een programma als een set opdrachten in een geformaliseerde programmeertaal gecodeerd, behoort typisch bij discrete computers. De steeds meer geavanceerde technologie zorgt ervoor, dat technische beperkingen om simulatiemodellen te bouwen ook minder stringent worden.

Was men in het verleden genoodzaakt om met grote computers te werken, nu zijn er software paketten verkrijgbaar waarmee op microcomputers gesimuleerd kan worden. Voorbeelden zijn 'micro-DYNAMO' (APPLE II en IBM-PC), 'SIMAX' (IBM-PC), 'micro-NET' (voor verschillende PC's bruikbaar) en SIMSCRIPT (IBM-PC). Verwacht wordt, dat deze ontwikkeling nog wel verder zal gaan. Het is ook mogelijk om met algemene talen zoals PASCAL of een goed gestructureerde BASIC (Microsoft of MSX) op microsystemen simulatiemodellen te ontwikkelen.

Het gunstige effect van de ontwikkelingen op het gebied van microcomputers is, dat simulatie uit de sfeer van een specialistische methode voor de 'happy few' kan komen. Hoewel voor meer complexe modellen met veel gegevens grotere computers noodzakelijk zijn, kunnen de micro's worden gebruikt voor kleine toepassingen en voor leerprocessen.

V.3 DE PROGRAMMATUUR

Naast de hardware is de software van groot belang voor computersimulatie. De voor dit doel beschikbare programmatuur is te verdelen in vier categorieën:

1. algemene programmeertalen, zoals Pascal
2. speciale simulatietalen, zoals SIMULA
3. simulatiepakketten, zoals DEMOS
4. statistische pakketten, zoals SPSS.

Aan elk van deze categorieën zal in het vervolg van dit hoofdstuk enige aandacht worden besteed. In deze paragraaf zullen we allereerst enkele voor simulatietoepassingen belangrijke kenmerken van programmeertalen behandelen. Vervolgens zal worden ingegaan op een aantal factoren, die de keuze van een programmeertaal kunnen beïnvloeden. De onderwerpen simulatiepakketten en statistische programmatuur worden summier behandeld in sectie 3. In de laatste sectie van deze paragraaf wordt iets gezegd over de manieren waarop het voortschrijven van de simulatietijd programmatisch kan worden behandeld.

V.3.1 Aan programmeertalen te stellen eisen

Zowel van simulatie als van programmeren wordt wel gezegd dat het een kunde is. Dit wordt onderstreept doordat boeken op beide terreinen soms in hun titel verwijzen naar de noodzaak van vakmanschap (skill). Als voorbeeld daarvan geven we de volgende twee titels.

Systems simulation, the art and science (R.E. Shannon)

The art of computer-programming (D.E. Knuth)

Echte kunde ofwel vakmanschap wordt gekenmerkt door het creatief en vakbekwaam hanteren van gereedschap en methoden uit het betreffende veld. In die zin vereist elke wetenschap kundigheid, dus ook simulatie en programmeren.

Programmering is een terrein dat een sterke ontwikkeling doormaakt. De oorzaken liggen voor de hand. Het is een nog jong gebied, maatschappelijk van groot belang en met grote financiële consequenties. In de loop van de jaren zijn honderden programmeertalen ontwikkeld. Veel daarvan zijn nauwelijks toegepast. Anderen, zoals COBOL, FORTRAN, BASIC en Pascal hebben een grote mate van populariteit weten te krijgen. Sommige talen hebben model gestaan voor een hele familie nieuwere talen. Een goed voorbeeld daarvan is ALGOL60. De afgeleide talen kunnen in het algemeen meer dan de moedertaal. Er zijn ook talen die zich door hun eigenschappen uitstekend lenen voor theoretische beschouwingen. Voorbeelden hiervan zijn ALGOL68 en SIMULA.

Sommige talen zijn speciaal gericht op specifieke toepassingen. Zo wordt Lisp sterk geassocieerd met kunstmatige intelligentie. Voor simulatie is een groot aantal specifieke talen ontwikkeld. Voorbeelden daarvan zijn SIMSCRIPT, GPSS, Dynamo. Ook de hierboven reeds genoemde taal SIMULA is oorspronkelijk als simulatietaal ontworpen. Wat bieden deze simulatietalen nu meer dan de eerder genoemde algemene talen? Is het nuttig om een taal als SIMULA te leren? Om deze vragen te kunnen beantwoorden, bespreken we enkele essentiële eisen waaraan een programmertaal moet voldoen om goed werkende simulatieprogramma's te kunnen ontwikkelen.

1. *De structuur van de taal moet goed gedefinieerd zijn.* De taal moet opgebouwd zijn volgens een voor de gebruiker toegankelijke formele syntaxis. Het is van belang dat zo'n taal dan 'transparant' voor de programmeur is. SIMULA en Pascal voldoen beide aan deze eis. Hiernaast moet ook de semantiek van de taal helder en goed beschreven zijn. De snelheid en het gemak waarmee een taal te leren valt, zijn hiervan sterk afhankelijk.

2. De taal moet faciliteiten bevatten om *de taak van de programmeur te verlichten.* Een uitgebreide verzameling mogelijkheden om data-structuren te kunnen definiëren en ermee te manipuleren is zonder meer noodzakelijk. Het 'class'-begrip bij SIMULA is een voorbeeld van een elegante flexibele data-structuur.

Voor simulatie is ook een tijdmechanisme of een 'klok' essentieel. In simulatietalen is een dergelijk mechanisme voorhanden; bij het gebruik van een algemene taal zal men zelf hiervoor procedures moeten ontwikkelen. Hierom zal aan het slot van deze paragraaf (§ V.3.4) iets nader op het tijdmechanisme worden ingegaan.

Wijzigingen zullen veelvuldig voorkomen; niet alleen bij het experimenteren, ook nà implementatie zal men het model willen aanpassen. De taal mag hier niet belemmerend optreden. Niet alleen zullen zich wijzigingen binnen de componenten voordoen, ook het aantal componenten zal moeten kunnen variëren. Fouten tegen een logische opbouw van het programma kunnen soms door de compiler gesignaleerd worden. Dat geeft een snelle 'feedback' naar de programmeur.

Tenslotte zal de beschrijving van de relaties tussen de delen onderling en tussen de delen en het geheel gemakkelijk en overzichtelijk beschreven moeten kunnen worden. Hiermee is een aantal faciliteiten genoemd, waarmee de ontwerper/programmeur in zijn werk ondersteund wordt ('het kan niet misgaan'); toch zal zelfs de 'beste' simulatietaal van een slecht opgezet model geen goed programma kunnen maken.

3. Voor simulatie met gebruik van kansvariabelen is *een toevalsgenerator* essentieel. Bij gebruik van een simulatietaal beschikt men over een toevalsgenerator en daarnaast over procedures om trekkingen te kunnen doen uit verschillende theoretische verdelingsfuncties, zoals de normale, de negatief exponentiële en de Poissonverdeling.

Een aantal algemene talen biedt tegenwoordig ook een standaard-procedure voor het trekken van toevalsgetallen. De mogelijkheid van trekkingen uit andere theoretische verdelingsfuncties ontbreekt meestal.

Terwijl aan de invoerkant een toevalsgenerator essentieel is, zijn ten behoeve van statistisch onderzoek van uitvoergegevens eenvoudige opslagmogelijkheden van belang. Deze gegevens kunnen dan met behulp van procedures uit een statistisch pakket worden bewerkt. De aanpassing van de uitvoer naar statistische routines zal eenvoudig moeten verlopen.

4. In veel simulatietoepassingen wordt met wachtrijen gewerkt. Een programmeertaal moet bij voorkeur over faciliteiten beschikken om hiermee om te kunnen gaan. SIMULA beschikt over een uitgebreid repertoire om *rijen (ketting- of lijststructuur)* te creëren. Manipulaties om elementen toe te voegen, te verwijderen, te bereiken en te veranderen zijn eenvoudige procedure-aanroepen. De administratie van deze datastructuren geschiedt intern.

5. Onontbeerlijk is vervolgens een krachtige *'foutenverklikker'*, niet alleen ten aanzien van syntaxfouten, maar ook ten aanzien van de logische opbouw van het programma en de wijze waarop de probleemspecificatie wordt ingebracht. Bij veel compilers laat dit onderdeel nogal te wensen over.

6. Naarmate simulatie minder een specialisme wordt en meer ten behoeve van toepassingen voor de besluitvorming van management wordt gebruikt, wordt de behoefte aan een *'eenvoudige'* taal groter. Met eenvoud bedoelen we nu dat men, zonder een 'echte' programmeur te zijn, snel zelf een model kan bouwen en een probleemspecificatie kan invoeren. Men beschrijft zijn probleem in een taal, die dicht bij een spreektaal ligt met een eenvoudige vocabulaire en syntax. Op die manier verkrijgt men een programma dat leesbaar blijft voor de materiedeskundige en dat dus als communicatiemiddel goed voldoet.

In de meest ideale situatie zou het op deze manier verkregen model interactief ingevoerd moeten kunnen worden om direct op fouten te worden getest. Na verbetering van de fouten kan vervolgens worden nagegaan of het model aan de verwachtingen voldoet.

Deze aanpak zal tot een grotere acceptatiegraad van het simulatiemodel kunnen leiden. Men zal sneller gebruik willen maken van simulatie als onderzoekmethode en als ondersteuning bij de besluitvorming. Men zal ook sneller geneigd zijn de resultaten van de simulatie-experimenten te vertrouwen.

Na deze opsomming van faciliteiten die een programmeertaal ten behoeve van simulatie-experimenten zou moeten bieden, bespreken we een viertal factoren die de keuze van de taal zullen beïnvloeden.

V.3.2 Keuzebepalende factoren

a. Eén van de belangrijkste factoren, in ieder geval de eerste keuzefactor, is de aard en de doelstelling van de probleemdefinitie. Hieruit volgt snel of een continu dan wel discreet proces nagebootst dient te worden. De keuze van een analoge, analoog-hybride of digitale rekenautomaat laten we verder rusten. Analoge simulatie van continue processen kan heel goed m.b.v. een digitale computer worden opgezet. Hiervoor zijn speciale simulatietalen beschikbaar. DYNAMO is hiervan een voorbeeld. Men moet hierbij denken aan simulatieprogramma's, waarbij de werking van een analoge computer wordt nagebootst.

Een andere mogelijkheid, hierbij aansluitend, is het bestaan van simulatietalen voor een specifiek vakgebied. Als voorbeeld valt te denken aan chemische of fysische (atoom- en kernfysica) processen of aan het ontwerpen van elektronische schakelingen (chips en micro's).

b. Een tweede categorie van keuzefactoren is gelegen in de uitgangspunten en de benaderingswijze van de probleemanalist. Wanneer de modellen wiskundig van aard zijn (bijvoorbeeld econometrische modellen), kunnen zij vaak beschreven worden volgens:

- een stelsel lineaire vergelijkingen
- een stelsel differentiaalvergelijkingen, of
- een stelsel differentievergelijkingen.

Hierbij dekt men ook vele vormen van continue processen. Er is een ruime keuze van simulatietalen voorhanden.

Niet altijd zijn de te simuleren systemen in mathematische formuleringen te beschrijven. Dan zal men in het programma de opdrachten noteren als oorzaak-gevolg relaties tussen de verschillende systeemcomponenten, of als een activiteitenstroom. Er is een verschil in benadering van deze problematiek te onderkennen in de verscheidene simulatietalen. Hierop komen we in § 4 terug.

c. Een derde verzameling keuzebepalende factoren komt voort uit de opvattingen van de analist/programmeur. Iemand die zich in één of meer algemene programmeertalen bekwaamd heeft, blijkt moeilijk te bewegen om een geheel nieuwe taal aan te leren voor een nieuw probleemgebied, zoals simulatie. Dit is één van de redenen waarom nog zolang in assembleertalen is geprogrammeerd, toen hogere talen reeds lang hun kinderziekten verloren hadden. Ook de hardnekkigheid, waarmee voor meer complexe toepassingen toch nog 'primitieve' talen als BASIC en FORTRAN gebruikt blijven worden en steeds uitgebreid worden om aan nieuwe eisen te kunnen blijven voldoen (BASIC+, FORTRAN IV, enzovoorts), is opvallend. De beweegreden om bij de bekende taal te blijven zal vaak liggen in de gedachte dat een simulatie-opdracht maar eenmaal voorkomt. In dat geval lijkt de investering om een geëigende simulatietaal te gebruiken wat overdreven. Wanneer evenwel de simulatie succesvol is, en waarom

zou dat niet het geval zijn, zal men ongetwijfeld nieuwe simulatie-opdrachten krijgen.

Het verdient de voorkeur om de mogelijke toepassingen waarbij simulatie gebruikt kan worden, eerst te inventariseren en dan een mede hierop afgestemde keuze van het gereedschap te maken.

d. De vierde keuzefactor is van technische en economische aard. Een aantal compilers voor simulatietalen stellen vrij hoge eisen aan de capaciteit van de hardware. Wanneer onvoldoende capaciteit beschikbaar is, wordt de keuze automatisch beperkt. Het is afhankelijk van de gewenste en geplande toepassingen of men bereid is grotere investeringen te doen. Een kosten/baten analyse zal moeten uitwijzen welke prijs men wil betalen voor de benodigde faciliteiten aan hard- en software en niet te vergeten mankracht. Deze drie zijn niet onafhankelijk en de aanschaf van computercapaciteit en een geschikte compiler zou een aanzienlijke besparing in mankracht met zich mee kunnen brengen. Ook de tijdsduur van het project kan positief worden beïnvloed.

V.3.3 Simulatiepakketten en statische programmatuur

Vooral voor mensen die incidenteel een simulatiemodel willen programmeren is het aanleren van een simulatietaal bezwaarlijk. Er gaat nogal wat energie zitten in het leren van zo'n taal. Gelukkig komen er tegenwoordig steeds meer *softwarepakketten* in de handel, waarmee het voor deze categorie gebruikers relatief eenvoudig is om een simulatiemodel te formuleren en er mee te experimenteren. Deze pakketten zijn in feite modelgeneratoren. Dit wil zeggen dat de pakketten de mogelijkheid bieden de specifieke eigenschappen en gegevens van het model toe te voegen aan een generatorprogramma. Met behulp van het aldus gecomplementeerde generatorprogramma kan de computer vervolgens een programma voor het model in kwestie samenstellen.

Een voorbeeld van zo'n pakket is DEMOS (afkorting voor Discrete Event Modelling On Simula). Dit pakket is in SIMULA geschreven. Voor de programmeur die DEMOS kent, is het niet noodzakelijk zich in SIMULA te verdiepen. De specifieke gegevens van een model worden als parameters toegevoegd aan DEMOS waarna dit programma door de SIMULA-vertaler kan worden vertaald. Immers DEMOS is in SIMULA geschreven.

De generatorprogramma's moeten niet worden verward met de in V.3.1 sub 6 bedoelde eenvoudige, dicht bij de spreektaal liggende talen. Hiervoor zijn zogenaamde interpreterende programma's nodig. Een interprator is een programma dat opdrachten interpreteert die in een andere taal zijn geschreven dan de machinetaal van de gebruikte computer.

De in het begin van deze paragraaf genoemde *vierde categorie van programmatuur* omvat de *statistische pakketten*. We doelen hierbij op een verzameling van statistische programma's die kunnen worden gebruikt om de uitvoergegevens van een simulatiemodel als invoergegevens te gebruiken. Op deze gegevens worden dan bewerkingen toegepast die door de gebruiker worden gewenst. Bijvoorbeeld het berekenen van een gemiddelde, de standaardafwijking of het uitvoeren van een analyse.

Voorbeelden van dergelijke pakketten zijn SPSS (Statistical Package for Social Sciences) en WESP (Waarlijk Eenvoudig Statistisch Pakket).

Samenvattend kan worden gesteld, dat de keuze van het juiste gereedschap niet eenvoudig is. Diverse elementen, waaronder heel persoonlijke spelen bij de keus een rol, zoals de beschikbaarheid van talen, de aard van de op te lossen problemen, de aanwezige talenkennis bij de programmeurs, de beschikbare tijd, de overdraagbaarheid van een model en de geschiktheid van een taal als communicatiemiddel.

Als eenmaal besloten is tot gebruik van een simulatietaal dan kunnen bij de keuze de volgende criteria worden gehanteerd*):

1. de beschikbaarheid van talen voor het aanwezige computersysteem
2. de kosten van aanschaf en onderhoud
3. het aantal simulatiestudies dat uitgevoerd zal worden
4. de soorten van systemen die gesimuleerd zullen worden
5. de kwaliteit van de documentatie
6. de moeilijkheidsgraad van de taal
7. de benodigde computercapaciteit
8. de verwerkingssnelheid van programma's
9. de flexibiliteit en de kracht van de taal.

V.3.4 Het tijdmechanisme in het simulatieprogramma

Er zijn globaal gesproken twee manieren waarop men het voortschrijden van de simulatietijd programmatisch kan organiseren: statisch of dynamisch. Men kan daartoe een vaste of een variabele waarnemingstijd kiezen.

Een vaste waarnemingstijd lijkt programmatisch wel voordelen te hebben, maar hierbij dienen we te bedenken, dat men de waarnemingsperiode niet zo maar willekeurig kan kiezen. Er mag eigenlijk per periode maar één toestandsverandering plaatsvinden, anders verliezen we de mogelijkheid tot controle en inzicht in het proces. Wil men dan ook nog met verdelingsfuncties en frequentieverdelingen

---

*) Zie R.E. Shannon, ibidem, pg. 107, 108.

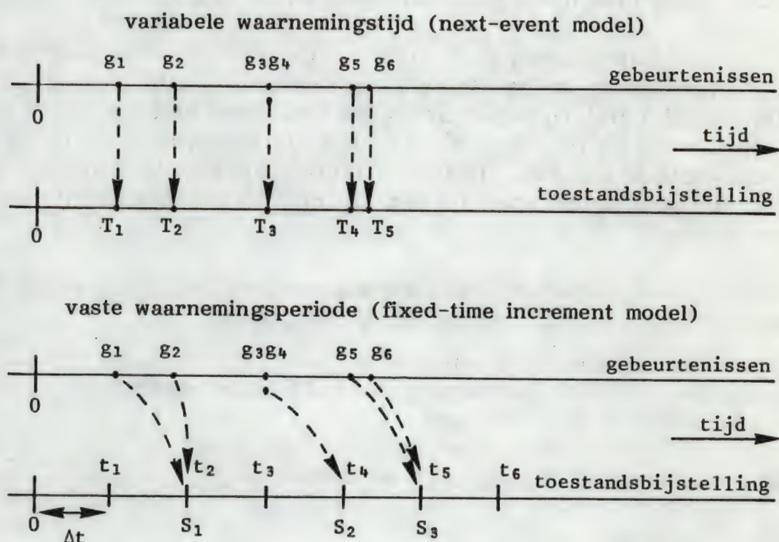
werken, dan zal de waarnemingsperiode als gevolg hiervan klein moeten zijn. Het logisch hieruit voortvloeiende nadeel is, dat het programma vele stappen (stapjes) in de tijd zal doen, zonder dat er iets gebeurt.

Het programma zal hierdoor bijzonder traag werken, vooral ook omdat alle punten in het programma, waar een gebeurtenis zou kunnen plaatsvinden, moeten worden bekeken.

Het alternatief voor een vaste waarnemingstijd is een variabele waarnemingstijd. Het systeem 'springt' van de ene gebeurtenis naar de andere en de tussentijd is programmatisch 'nihil'.

Het simulatiesysteem dient door de taal zelf geheel bestuurd en geadministreerd te worden zonder (veel) bemoeienis van de programmeur. Daartoe zijn bepaalde procedures aanwezig om de simulatietijd te beheersen (een interne 'klok'), gebeurtenissen te signaleren, toestandsvariabelen bij te werken, speciale testcondities te herkennen en is er een flexibel datasysteem om de genoemde zaken te registreren en administreren.

Zie voor een nadere toelichting figuur V.3a.



In het geval van een variabele waarnemingstijd worden de toestandsvariabelen bijgewerkt op het moment dat er echt iets gebeurt. De samenvallende gebeurtenissen g_3 en g_4 worden natuurlijk door het programma wel serieel verwerkt; een adequaat besturingsmechanisme voorziet in een goede verwerking.

Bij de vaste waarnemingsperiode wordt na elk tijdsinterval Δt nagegaan of één of meer gebeurtenissen zijn opgetreden in het afgelopen tijdsinterval. Op tijdstip t_1 leidt dit onderzoek tot een vergeefse poging, terwijl op het volgende tijdstip t_2 twee gebeurtenissen g_1 en g_2 moeten worden verwerkt. Hetzelfde is het geval met de gebeurtenissen g_3 en g_4 (uiteraard) en g_5 en g_6. In feite is hier het interval Δt al te groot gekozen.

Figuur V.3a

V.4 BENADERINGSWIJZEN VOOR HET FORMULEREN VAN SIMULATIEPROCESSEN

In de vorige paragraaf is een aantal voor de keuze van een programmeertaal bepalende factoren besproken. Als een belangrijke factor is genoemd de benaderingswijze van het model door de probleemanalist. Er kunnen bij discrete simulatie drie benaderingswijzen voor het beschrijven van systemen worden onderscheiden:

a. de gebeurtenissen-beschrijvingsmethode
b. de activiteiten-beschrijvingsmethode
c. het proces-beschrijvingsmethode.

Bij het ontwerpen van simulatietalen is meestal één van deze drie methoden als uitgangspunt gekozen, zodat gesproken wordt van gebeurtenis-georiënteerde (Engels: event-oriented) talen en proces-georiënteerde (Engels: process-oriented) talen. We zullen aan elk van deze beschrijvingsmethoden enige aandacht schenken.

a. De *'event' benadering* gaat uit van gebeurtenissen (die een toestandsverandering in het systeem genereren), die zo gepland zijn in de tijd dat het tijdsbesturingsmechanisme aan de hand van zekere condities kan bepalen of zo'n gebeurtenis plaatsvindt. De gebeurtenis zelf is tijdloos. Het besturingssysteem 'houdt de tijd bij' en zet de gebeurtenissen in een tijdrij (schedule) zodat de volgorde vastligt.

Voorbeeld: In een loketsysteem (zie bijvoorbeeld § IV.5) kunnen als gebeurtenissen worden aangemerkt:

- aankomst van een klant
- de start van de bediening door de loketbeambte
- het vertrek van de klant.

De gebeurtenissenlijst kan er dan zo uitzien:

tijd	*gebeurtenis*
0	loket open
3	aankomst 1° klant
3	1° klant wordt geholpen
4	aankomst 2° klant (gaat in de wachtrij)
6	aankomst 3° klant (gaat in de wachtrij)
8	vertrek 1° klant
8	2° klant wordt geholpen (uit de rij verwijderd)
12	aankomst 4° klant (gaat in de wachtrij)
enz.	

Wanneer men zelf een klokmechanisme maakt met een waarnemingsperiode van 1 minuut, dan is bovenstaand model gemakkelijk in een algemene programmeertaal op te zetten, zij het dat we dan niet een bijzonder flexibel model hebben.

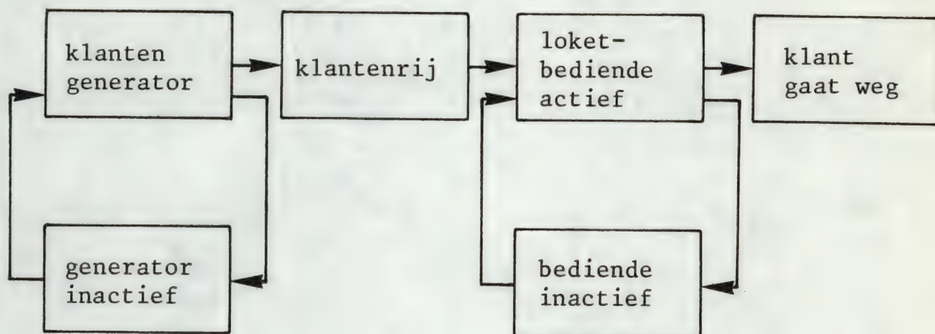
Ga zelf eens na welk gevolg de introductie van meer loketten en van meer bedienden, of van een andere rijdiscipline zou hebben.

Voorbeelden van 'event'-georiënteerde simulatietalen zijn: SIMSCRIPT en GASP.

b. Bij *activiteit-georiënteerde beschrijvingen* worden tijdsafhankelijke handelingen als momentane (tijdloze) gebeurtenissen gerepresenteerd. Voordat de simulatietijd in het programma over de tijdas verschoven wordt, wordt door een uitvoeringsbesturingsprocedure de verzameling condities afgelopen, om te bepalen welke handelingen plaatsvinden. Elke handeling wordt beschreven door een eigen procedure. Om zich deze werkwijze te kunnen voorstellen, denkt men zich in toeschouwer te zijn en als zodanig naar het systeem te kijken. Probeer dan de activiteiten te onderscheiden als de verzamelingen acties die de toestandsvariabelen op gelijke wijze doen veranderen.

Voorbeelden van activiteit-georiënteerde talen zijn CSL, ESP en FORSIM-IV.

Het loketsysteem toont het volgende beeld:



activiteiten:

- plaats klant in de rij; - bepaal nieuw aankomstmoment	- bepaal moment dat bediening beëindigt	- klant verlaat het systeem

condities:

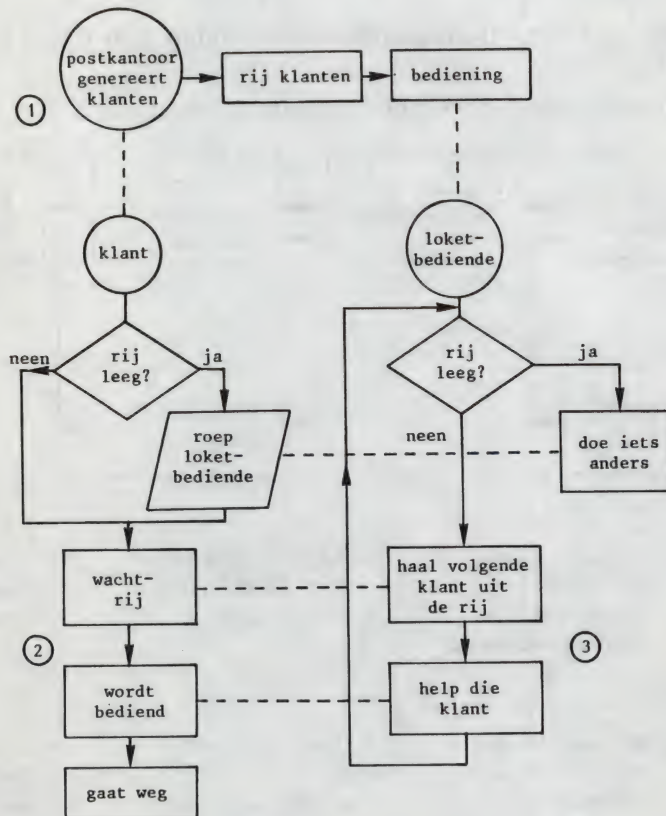
- aankomstmoment van een klant is bereikt	- loket is vrij; - de rij is niet leeg	- eindtijdstip bediening is bereikt

Figuur V.4a

c. *Proces-georiënteerde beschrijvingsmethoden* geven qua filosofie een *mengvorm* van activiteiten en gebeurtenissen weer. Een proces is te definiëren als een vaste verzameling van activiteiten en gebeurtenissen. Een proces kent een dynamisch en wederkerend gedrag. Processen zijn flexibel, kunnen in een tijdrij gezet worden (Engels: scheduling), zijn te onderbreken, kunnen in deelprocessen onderverdeeld worden, zijn in de tijd te verschuiven, enzovoorts. In een programma kunnen de processen beschouwd worden als onafhankelijke deelprogramma's, waarin zowel handelingen als gebeurtenissen hun plaats hebben. Het besturingsmechanisme houdt bij welke processen actief zijn, waar herstartende processen gebleven zijn en wat de volgorde van de te activeren processen is.

Voorbeelden van proces-georiënteerde talen zijn SIMULA, OPS en SOL. Een proces-georiënteerd simulatieprobleem zal meestal een specifieke procesgerichte taal vereisen, waarbij een mechanisme van de 'co-routine' aanwezig is. Een taal als Pascal-Plus of parallel Pascal voldoet ook aan deze eis.

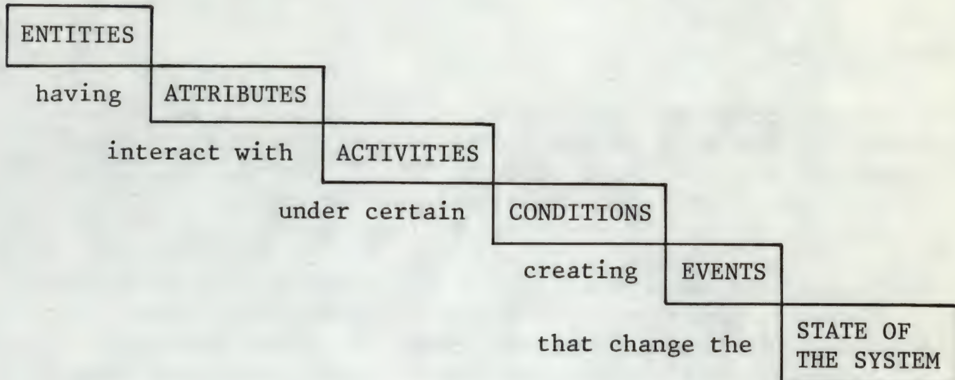
Het loketvoorbeeld kan als volgt in drie processen worden uitgebeeld. De stippellijnen geven de relaties tussen de processen aan.



Figuur V.4b

Vaak zal de aard van het simulatieprobleem wel een zekere aanwijzing geven omtrent voorkeur voor een van de drie benaderingswijzen. Het is ook mogelijk om in een proces-georiënteerde taal, zoals bijvoorbeeld SIMULA, de event- of activity-benadering toe te passen. Om deze reden moet men het belang van het onderscheid in de drie benaderingswijzen ook niet overschatten. Vooral het verschil tussen de activiteitenbenadering en de procesbenadering is principieel niet bijzonder groot.

We besluiten dit overzicht met een figuur uit Shannon*).



Figuur V.4c

*) Zie: R.E. Shannon, ibidem, pg. 109.

V.5 BASIC EN PASCAL

In deze paragraaf zullen we enige aandacht besteden aan twee, vooral in het onderwijs veel gebruikte programmeertalen: BASIC en Pascal. Beide talen zijn niet voor simulatietoepassingen ontwikkeld. Ze bieden daartoe dan ook beperkte mogelijkheden, Pascal overigens meer dan BASIC. Naast enkele algemene opmerkingen over deze talen zal van elk een programmavoorbeeld worden gegeven.

V.5.1 BASIC

De programmeertaal BASIC heeft zich de laatste jaren zowel ten aanzien van de structuur als ten aanzien van de mogelijkheden in positieve zin ontwikkeld. Moderne BASIC-versies, zoals Microsoft BASIC*) bieden de mogelijkheid goed gestructureerde programma's te schrijven. De programmeur kan bij gebruik van deze versies over 'if ..... then ..... else .....' en 'while ..... wend' opdrachten beschikken. Het gebruik van GOTO kan zodoende worden vermeden. Ook in andere opzichten biedt de taal zeker mogelijkheden om een programma doorzichtig en gestructureerd in elkaar te zetten.

Helaas beschikken ook de moderne BASIC-versies niet over de speciale toevoegingen, waarmee simulatieprogramma's prettiger en sneller kunnen worden ontworpen. We doelen in dit verband op mogelijkheden als:

- het genereren van kansvariabelen uit specifieke verdelingsfuncties,
- het bijhouden van gebeurtenissen in de tijd,
- het toevoegen van records aan, of weghalen van records uit een lijst en
- het verzamelen en analyseren van gegevens.

BASIC is dus beperkt. Toch is het ook in BASIC mogelijk eenvoudige simulatietoepassingen te programmeren. Het voorbeeld van de haringkar laat zich redelijk gemakkelijk in BASIC uitwerken.

Voorbeeld simulatieprogramma in BASIC

---

*) Zie bijvoorbeeld het boek 'Programmeercursus Microsoft BASIC' door Nok van Veen, Academic Service, 1983.

Voorbeeld simulatieprogramma in BASIC

```
10      REM  Het haringkar programma in MBASIC
20      DIM WACHTRIJ(30)
30      DEF FNNEGEXP( AANKOMSTGEMIDDELDE)= -LOG(RND)/AANKOMSTGEMIDDELDE
40      DEF FNUNIFORM (A,B) = A + (B - A) * RND
50      INPUT A, B, AANKOMSTGEMIDDELDE, SIMULATIETIJD
51      LAATST = 0
52      AANTALKLANTEN = 0
53      AANKOMSTTIJD = 0
54      VISMAN = 0
55      WACHTTIJD = 0
56      SOMWACHTTIJD = 0
60      WHILE AANKOMSTTIJD < SIMULATIETIJD
70         AANTALKLANTEN = AANTALKLANTEN + 1
80         LAATST = LAATST + 1
90         AANKOMSTTIJD = AANKOMSTTIJD + FNNEGEXP (AANKOMSTGEMIDDELDE)
100        WACHTRIJ(LAATST) = AANKOMSTTIJD
110        WHILE VISMAN <= AANKOMSTTIJD
120        WACHTTIJD = VISMAN - WACHTRIJ(1)
130           IF WACHTTIJD < 0 THEN WACHTTIJD = 0 :
                                   VISMAN = WACHTRIJ(1) + FNUNIFORM(A,B)
                              ELSE VISMAN = VISMAN + FNUNIFORM(A,B)
140           SOMWACHTTIJD = SOMWACHTTIJD + WACHTTIJD
150           FOR N = 1 TO LAATST
160           WACHTRIJ(N) = WACHTRIJ (N+1)
170           NEXT N
180           LAATST = LAATST - 1
190        WEND
200     WEND
209     LPRINT "de invoer: A=";A;";B=";B;";aankgem=";AANKOMSTGEMIDDELDE
210     LPRINT "de simulatietijd is: "; SIMULATIETIJD
220     LPRINT "het aantal klanten is: "; AANTALKLANTEN
230     LPRINT "de totale wachttijd is: "; SOMWACHTTIJD
240     LPRINT "de gemiddelde wachttijd is: ";SOMWACHTTIJD/AANTALKLANTEN
250     END
```

```
de invoer: A= 2 ;B= 2 ;aankgem= .5
de simulatietijd is:  4
het aantal klanten is:  1
de totale wachttijd is:  0
de gemiddelde wachttijd is:  0

de invoer: A= 2 ;B= 2 ;aankgem= 2
de simulatietijd is:  4
het aantal klanten is:  7
de totale wachttijd is:  1.929714
de gemiddelde wachttijd is:  .2756734

de invoer: A= 0 ;B= 2 ;aankgem= .5
de simulatietijd is:  10
het aantal klanten is:  8
de totale wachttijd is:  5.639245
de gemiddelde wachttijd is:  .7049056

de invoer: A= 0 ;B= 2 ;aankgem= 2
de simulatietijd is:  10
het aantal klanten is:  24
de totale wachttijd is:  20.69132
de gemiddelde wachttijd is:  .8621381
```

V.5.2 Pascal

Pascal is een 'jonge' taal die in betrekkelijk korte tijd zeer populair is geworden. Ze wordt in het onderwijs in ruime mate gebruikt en is op veel microcomputers beschikbaar. In Nederland heeft de Stichting Teleac een televisiecursus aan Pascal gewijd waardoor de taal ruime bekendheid heeft gekregen.

Ten opzichte van BASIC heeft Pascal het nadeel dat het meer inspanning kost om haar (volledig) te leren. Daar staat tegenover, dat men dan ook een taal kent die meer mogelijkheden biedt. Pascal is namelijk een taal die voor een breed scala van toepassingen kan worden gebruikt. Ze beschikt over goede programmeer- en datastructuren en levert, bijna per definitie overzichtelijke, goed leesbare programma's. Juist als programma's groter en complexer worden, gaan de voordelen van Pascal ten opzichte van BASIC (en ook van andere algemene programmeertalen) sterk overheersen.

Voor het aanleren van een goede programmeerattitude, hetgeen in onderwijs van groot belang is, is Pascal zeer geschikt. Bovendien blijkt in de onderwijspraktijk dat kennis van Pascal een goede basis is voor het leren van een simulatietaal als SIMULA (zie § V.6).

Niet alleen de voor goede programma's noodzakelijke structuren als *if .... then .... else ...., while .... do, repeat .... until* en *case,* maar ook de mogelijkheid om zelf andere types als *real, integer* en *char* te definiëren alsmede de datastructurerings-mogelijkheden maken Pascal ook redelijk geschikt voor simulatie-doeleinden. Een voorbeeld kan deze uitspraak verduidelijken.

```
type gebeurtenis = (aankomst, bediening, opening, sluiting);
```

Als we nu declareren var volgende gebeurtenis: gebeurtenis; dan kan de programmastructuur deze vorm hebben:

```
repeat get (volgende gebeurtenis);
     case volgende gebeurtenis of
        aankomst:  .......;
        bediening: .......;
        opening:   .......;
        sluiting:  .......;
     end
until volgende gebeurtenis = sluiting;
```

Op de stippellijnen staan opdrachten - gewoonlijk procedure-aanroepen - die behoren bij de akties van de respectievelijke gebeurtenissen. Als dus de volgende gebeurtenis het starten van het bedieningsproces blijkt te zijn, worden de akties achter *bediening:* verwerkt.

Een ander voorbeeld laat zien hoe we de verzameling eigenschappen van een object in een 'record' kunnen definiëren.

```
type klant = record
             aankomsttijd, helptijd: real;
             aantal boodschappen: integer;
             .
             .
             .
             end;
```

Aan het slot van deze paragraaf geven we twee volledige simulatieprogramma's in Pascal als voorbeeld.

De haringkar met één bediende

```
        PROGRAM haring (input,output);
VAR aankomstgemiddelde, simulatietijd, visman, wachttijd, somwachttijd,
    aankomsttijd: real;
    a, b, n, laatst, aantalklanten: integer;
    wachtrij: ARRAY[0..30] of real;
FUNCTION negexp( aankomstgemiddelde: real): real;
        BEGIN negexp := -ln(random)/aankomstgemiddelde; END;
FUNCTION uniform(a, b: integer): real;
        BEGIN uniform := a + (b-a)*random; END;

BEGIN
    read (a, b, aankomstgemiddelde, simulatietijd);
    laatst := 0; aantalklanten :=0; aankomsttijd :=0;
    visman :=0; wachttijd :=0; somwachttijd:=0;
    WHILE aankomsttijd < simulatietijd DO
    BEGIN
        aantalklanten := aantalklanten + 1;
        laatst := laatst + 1;
        aankomsttijd := aankomsttijd + negexp(aankomstgemiddelde);
        wachtrij[laatst] := aankomsttijd;
        WHILE visman <= aankomsttijd DO
        BEGIN
            wachttijd := visman - wachtrij[1];
            IF wachttijd < 0
               THEN BEGIN
                    wachttijd := 0;
                    visman := wachtrij[1] + uniform(a, b);
                    END
               ELSE
                    visman := visman + uniform(a, b);
            somwachttijd := somwachttijd + wachttijd;
            FOR n := 1 TO laatst DO
                wachtrij[n] := wachtrij[n+1];
            laatst := laatst - 1;
        END;
    END;
    writeln('de simulatietijd is: ',simulatietijd);
    writeln('het aantal klanten is: ',aantalklanten);
    writeln('de totale wachttijd is: ',somwachttijd);
    writeln('de gemiddelde wachttijd is: ',somwachttijd/aantalklanten);

    (* In dit PASCAL programma is geen gebruik gemaakt van bijzondere PASCAL
    mogelijkheden zoals pointers en types. De variabele VISMAN bevat de tijd
    waarop de visman klaar is met het helpen van een klant. De array WACHTRIJ
    bevat de aankomsttijden van de klanten die nog niet geholpen zijn. Als de
    visman vrij is, helpt hij de eerste klant uit de rij. De variabele
    AANKOMSTTIJD dient als 'klok'. De uitvoer is ook primitief. Een meer uit-
    gebreide uitvoer zal meer relevante gegevens kunnen produceren. *)
    END.
```

Voorbeeld simulatieprogramma in Pascal: De haringkar met méér bedienden.

```
PROGRAM haring2 (input,output);
(* In dit programmavoorbeeld zijn meer bedienden actief *)

VAR aankomstgemiddelde, simulatietijd, wachttijd, somwachttijd,
    aankomsttijd: real;
    a, b, n, laatst, aantalklanten, eersteklaar, aantalbedienden: integer;
    wachtrij: ARRAY[0..30] of real;
    bediende: ARRAY[0..5]  of real;
FUNCTION negexp( aankomstgemiddelde: real): real;
        BEGIN negexp := -ln(random)/aankomstgemiddelde; END;
FUNCTION uniform(a, b: integer): real;
        BEGIN uniform := a + (b-a)*random; END;

BEGIN
    read (a, b, aankomstgemiddelde, simulatietijd, aantalbedienden);
    laatst := 0; aantalklanten := 0; aankomsttijd := 0;
    wachttijd := 0; somwachttijd := 0; eersteklaar := 1;
    FOR n := 0 TO 5 DO bediende[n] := 0;
    WHILE aankomsttijd < simulatietijd DO
    BEGIN
        aantalklanten := aantalklanten + 1;
        laatst := laatst + 1;
        aankomsttijd := aankomsttijd + negexp(aankomstgemiddelde);
        wachtrij[laatst] := aankomsttijd;
        WHILE (bediende[eersteklaar] <= aankomsttijd) AND (laatst > 0) DO
        BEGIN
            wachttijd := bediende[eersteklaar] - wachtrij[1];
            IF wachttijd < 0
               THEN BEGIN
                    wachttijd := 0;
                    bediende[eersteklaar] := wachtrij[1] + uniform(a, b);
                    END
               ELSE
                    bediende[eersteklaar] := bediende[eersteklaar]
                                             + uniform(a, b);
            somwachttijd := somwachttijd + wachttijd;
            FOR n := 1 TO laatst DO
                wachtrij[n] := wachtrij[n+1];
            laatst := laatst - 1;
            (* hier wordt iedereen in de rij een plaats opgeschoven *)
            FOR n := 1 TO aantalbedienden DO
                    IF bediende[eersteklaar] > bediende[n]
                       THEN eersteklaar := n;
                    (* nu wordt gekeken welke verkoper het eerste klaar is *)
        END;
    END;
    writeln('de simulatietijd is: ',simulatietijd);
    writeln('het aantal klanten is: ',aantalklanten);
    writeln('de totale wachttijd is: ',somwachttijd);
    writeln('de gemiddelde wachttijd is: ',somwachttijd/aantalklanten);

    (* In dit PASCAL programma is evenmin gebruik gemaakt van bijzondere PASCAL
    mogelijkheden zoals pointers en types. De ARRAY "bediende" bevat de tijd
    waarop de verkopers klaar zijn met het helpen van een klant. De array
    WACHTRIJ bevat de aankomsttijden van de klanten die nog niet geholpen zijn.
    Als een verkoper vrij is, helpt hij de eerste klant uit de rij. De variabele
    AANKOMSTTIJD dient als 'klok'. De uitvoer is ook hier te primitief.
    Een meer uitgebreide uitvoer zal meer relevante gegevens moeten produceren.
END.
```

V.6 SIMULA: EEN SIMULATIETAAL

SIMULA 67 is ontworpen in Noorwegen als uitbreiding van ALGOL 60 ten behoeve van simulatieprogramma's. De taal is een krachtig en flexibel simulatiegereedschap en kan als een succes worden bestempeld. De kracht van SIMULA is gelegen in de bijzondere mogelijkheden ten behoeve van systeembeschrijving en modelformulering. Vooral ten aanzien van het ontwikkelen van informatiesystemen heeft SIMULA zijn gebruikswaarde ten volle getoond. Als voorbeelden hiervan noemen we systeembeschrijvingen voor de activiteiten van systeemanalyse, generering van alternatieve systeemontwerpen en database systemen.

Het centrale concept in SIMULA is de class. Een class lijkt op een procedure, maar er zijn enkele kenmerkende verschillen:

1. terwijl van een procedure tegelijk slechts één ingevuld exemplaar aanwezig is, kunnen van een class meerdere ingevulde exemplaren (objecten) parallel bestaan;

2. de variabelen in een procedure zijn lokaal en kunnen in de rest van het programma niet gebruikt worden, de lokale namen van een class kunnen via een speciale notatie benaderd worden.

Aan de hand van een voorbeeld uit Birtwistle*) laten we zien hoe een 'class' gedeclareerd wordt en op welke wijze objecten gegenereerd kunnen worden.

We maken een verzameling voertuigen. Alle hebben als attributen: bouwjaar en gewicht. Bovendien willen we kunnen aangeven of een auto defect is. De declaratie ziet er als volgt uit:

```
class voertuig (bouwjaar, gewicht);
      integer bouwjaar, gewicht;
  begin boolean defect;
        if bouwjaar < 1885 or gewicht ≤ 0
           then genereer een foutboodschap;
  end voertuig;
```

Er is nu nog geen enkel voertuig, deze moeten nog 'gefabriceerd' worden. Daarvoor is een referentie nodig: een naam, waarmee straks naar een gecreëerd object verwezen kan worden.

---

*) G.M. Birtwistle, Discrete event modelling on SIMULA, London, 1979.

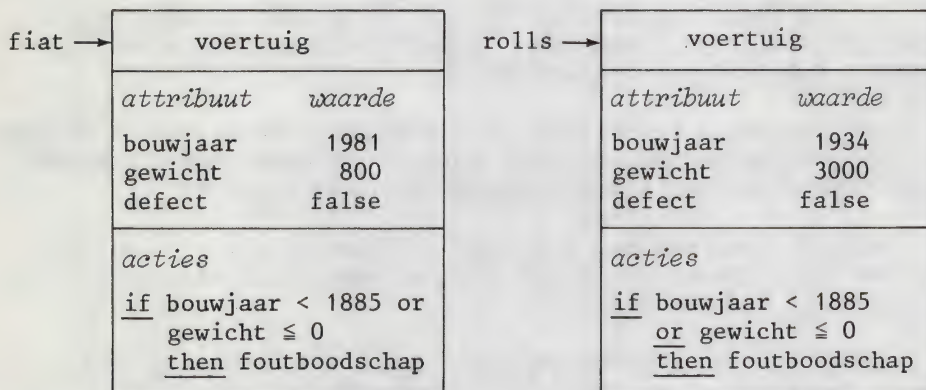
```
ref(voertuig) fiat, rolls;
```

Potentieel is nu een tweetal voertuigen benoemd, maar er is nog steeds geen object: de waarde van fiat en rolls is none.

De creatie tenslotte gebeurt als volgt:

```
fiat :- new voertuig(1981, 800);
rolls :- new voertuig(1934, 3000);
```

Van beide objecten valt het bestaan te visualiseren met een plaatje, waarin de eigenschappen (attributes) en acties (actions) duidelijk naar voren komen.



In de actielijst van objecten kunnen alle vormen van SIMULA-opdrachten voorkomen, óók procedures. De acties worden in principe uitgevoerd bij de creatie van het object.

In het programma zijn de attributen via de zogenaamde 'dot'-notatie gemakkelijk bereikbaar:

```
if rolls. defect
   then outtext('treurmuziek')
   else stap in en rij weg;

if botsing
   then fiat. defect := true;
```

Door middel van 'prefixing' kan men een hiërarchie van 'classes' opbouwen. Veronderstel, dat men een onderscheid wil maken tussen vrachtwagens, personenauto's, autobussen en kraanwagens. Voor elke categorie geldt dat alle objecten de drie attributen bouwjaar,

gewicht en defect gemeen hebben. Het levert veel overbodig werk om dit bij elke definitie van een 'class' separaat op te nemen. Dit wordt gedemonstreerd voor de vrachtwagen:

```
class vrachtwagen( bouwjaar, gewicht, laadvermogen);
      integer bouwjaar, gewicht, laadvermogen;
   begin boolean defect, wegreus;
         real lading;
         if bouwjaar < 1885 or gewicht ≤ 0
            then foutboodschap;
         wegreus := laadvermogen > 2500;
   end vrachtwagen;
```

Met gebruikmaking van de 'prefix-class' voertuig krijgen we evenwel de volgende declaratie:

```
voertuig class vrachtwagen(laadvermogen);
               integer laadvermogen;
         begin boolean wegreus;
               real lading;
               wegreus := laadvermogen > 25000;
         end vrachtwagen;
```

Op deze wijze kan men 'classes' definiëren, die bijvoorbeeld een geheel verkeerssysteem definiëren. Hiervan is dan standaard gebruik te maken. In een zelf-gedefinieerd programma kan men het verkeer manipuleren, gebruikmakend van de 'class' verkeer uit een programma bibliotheek.

```
begin external class verkeer;
   verkeer
      begin
            eigen declaraties;
            eigen opdrachten; objecten en attributen
            uit de class verkeer mogen worden gebruikt;
      end;
end;
```



De relaties in een simulatiemodel zijn gemakkelijk in te voeren. Men kan namelijk in een object procedures of opdrachten als attribuut opnemen, die op andere objecten inwerken.

Zoals de 'class' verkeer tot een standaard-'class' te maken is, zo heeft SIMULA zelf systeemgedefinieerde 'classes'. Ten behoeve van simulatie zijn de twee belangrijkste 'classes' SIMSET en SIMULATION.

SIMSET is bedoeld om met tijden te kunnen manipuleren (circular list en two-way list).

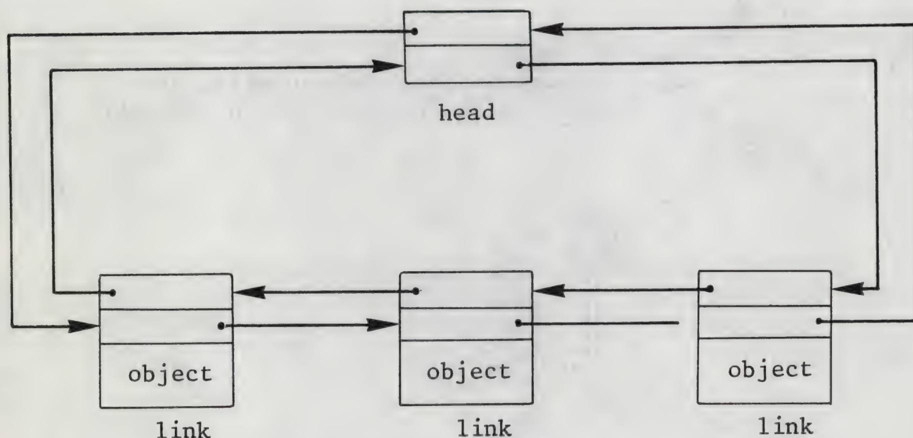
SIMSET beschikt over procedures voor de volgende doeleinden:

- het geven van verwijzingen naar de voorganger of de opvolger van een object in een verzameling,
- het verwijderen van een object uit een lijst of verzameling,
- het invoegen van een object achterin de lijst,
- het invoegen van een object voor of achter een omschreven ander object,
- het bepalen of een verzameling leeg is,
- het leegmaken van een verzameling,
- het bepalen van het aantal objecten in een verzameling,
- het geven van een verwijzing naar het laatste object in een lijst.

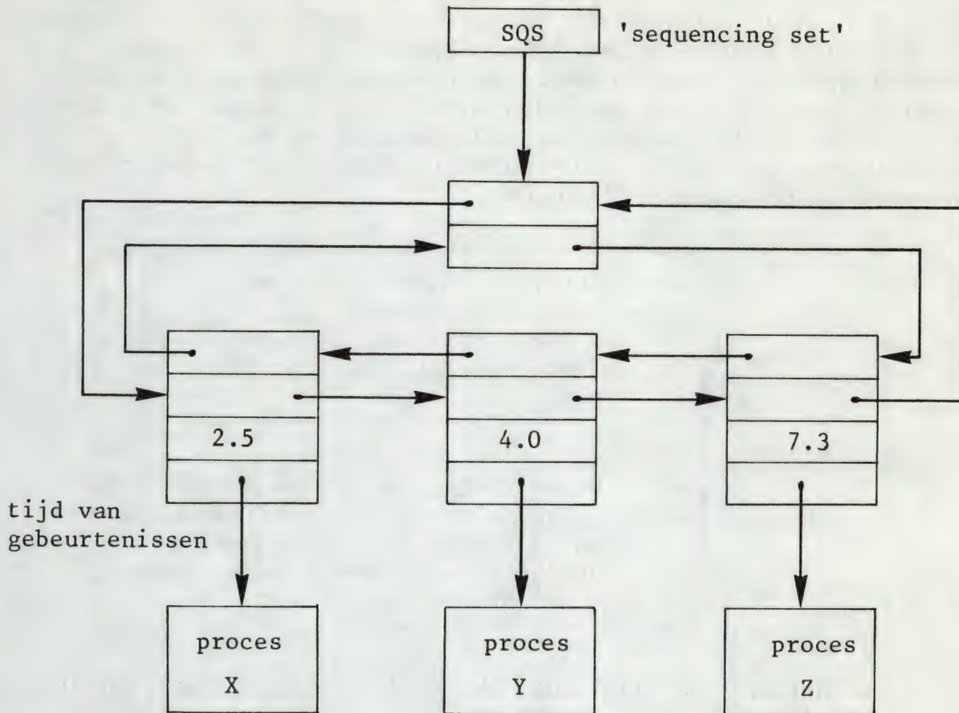
Men kan eenvoudig over deze procedures beschikken door het eigen programma te prefixen met 'SIMSET'.

De 'class' SIMULATION dient om te kunnen manipuleren met objecten in de tijd. Tijdstippen van executie, (tijdelijk) stoppen, (her-)starten, enzovoorts kunnen toegewezen worden aan de objecten in het systeem. Daar SIMULATION als prefix-class ook SIMSET bevat, kan men met de prefix SIMULATION automatisch over de rij-manipulatie procedures beschikken.

De structuur van de rij in SIMSET is als volgt te schetsen.



De 'class' SIMULATION gebruikt SIMSET om de tijd als een rij processen te organiseren. Intern wordt bijgehouden wanneer een bepaalde gebeurtenis moet plaatsvinden. Het proces waarin deze gebeurtenis is ondergebracht wordt dan 'actief'. In beeld:



In de rij SQS worden de tijden waarop gebeurtenissen plaatsvinden in volgorde van voorkomen bijgehouden. Vanuit de elementen in deze rij verwijzen pointers naar die gebeurtenissen, die in de vorm van processen in het SIMULA-programma zijn opgenomen. Het eerst aan de beurt zijnde proces is proces X. De 'simulatieklok' krijgt de tijd waarop dit proces is 'gepland'. De klok blijft op die tijd (2.5) staan totdat het proces X niet verder verwerkt wordt (het proces is beëindigd of wordt opgeschort). Daarna wordt de klok op 4.0 gezet en start proces Y. Het is echter mogelijk dat binnen proces X activiteiten hebben plaatsgevonden waardoor een ander proces, bijvoorbeeld Q, eerder dan proces Y 'actief' moet worden. Dit proces Q wordt dan door het in SIMULA ingebouwde tijdmechanisme vóór proces Y in de rij SQS geplaatst.

Binnen SIMULATION is een aantal procedures en opdrachten aanwezig waarmee de programmeur de volgorde van gebeurtenissen kan regelen en eveneens 'quasi-parallelle' gebeurtenissen kan doen plaatsvinden.

V.7 VOORBEELD SIMULATIEPROGRAMMA IN SIMULA

In een eerste opzet gaan we uit van het eenvoudige model van de haringkar, zoals dat beschreven is in § I.3. Er is één verkoper die de klanten in volgorde van aankomst (volgens het principe 'first in, first out') helpt. Als het systeem leeg is, activeert een aankomende klant de verkoper door een bestelling te doen. Klant en verkoper zijn dan zolang de bedieningstijd duurt 'bezet'. We zullen nu de acties van de systeemcomponenten (klant en verkoper) catalogiseren, waarbij het dan vanzelf duidelijk wordt welke relaties er tussen de componenten bestaan.

de klant:

```
1  Klant arriveert in het systeem
2  als de verkoper bezig is
        dan wacht hij op zijn beurt
        anders geeft hij een bestelling op;
```

de verkoper:

```
1  Staan er klanten te wachten?
2  niet: wacht rustig tot iemand komt;
3  wel:  vraag de aan de beurt zijnde klant
         wat deze nodig heeft en bedien
         hem/haar gedurende de helptijd;
4  Herhaal het proces 1-3;
```

Voor het systeem geldt nog: als een klant binnenkomt, wordt deze in de rij geplaatst en als de verkoper de aan de beurt zijnde klant gaat helpen, wordt deze uit de rij gehaald*).

Om nu tot een beschrijving in SIMULA te komen, beschrijven we nog een aantal belangrijke punten.

- De haringkar wordt gesimuleerd om te zien of een tweede verkoper zinvol zou zijn. Willen we dat weten, dan is er informatie nodig over de wachttijden, met name de gemiddelde wachttijd. Ook de variantie en het maximum van de wachttijd zijn interessante gegevens. Daarnaast is het mogelijk om het rijlengte-patroon te achterhalen, benevens de tijd waarin de verkoper niets doet ('idle time').

- De resultaten zullen in belangrijke mate afhangen van de modelparameters. Deze zijn in ons geval:

---

*) Dit is niet de enige mogelijkheid het model te beschrijven. Het is in het algemeen aan te bevelen meerdere beschrijvingen uit te proberen.

- het aankomstpatroon van de klanten
- de bedieningstijd van de verkoper

Teneinde flexibel te kunnen experimenteren, worden deze parameters bij het begin van de simulatie ingelezen.

- Zowel de klant als de verkoper ondernemen acties in de tijd, die de andere systeemcomponent(en) beïnvloeden. Beide objecten zijn leden van 'process-classes'. Een object van een 'process-class' kan in de volgende toestanden verkeren:
  - actief (een klant die zich in de rij plaatst);
  - opgeschort (de verkoper is gedurende de helptijd niet voor iets anders beschikbaar);
  - passief (een klant in de rij wacht lijdzaam totdat hij door de verkoper wordt aangesproken);
  - klaar (de geholpen klant, die na betaling vertrekt);

De real procedure 'time' houdt de simulatiesysteemtijd bij. Time geeft de tijd (het moment) aan van een actief object. De tijdsduur, dat een object actief is, is in het *programma* altijd nul.

De procedures, die op de toestand van de objecten werken en die wij gebruiken (zie een SIMULA handleiding voor een complete taalbeschrijving) zijn de volgende:

hold (tijdsduur): Het object wordt gedurende 'tijdsduur' opgeschort. Na afloop wordt het object weer actief, namelijk als de tijdsduur voorbij is.

passivate: Het object, van waaruit de procedure passivate wordt aangeroepen, wordt passief gemaakt.

activate, reactivate: Een passief object wordt actief gemaakt.

Nu kunnen we een *modelbeschrijving* geven in 'pseudo-SIMULA', waarbij de details nog niet zijn uitgewerkt, maar de structuur van het programma helder en duidelijk naar voren komt.

```
process class klant(tijd); real tijd;
   begin ga in de rij staan;
         noteer de aankomsttijd;
         if de verkoper niets doet
            then activate verkoper
            else passivate;
         noteer de wachttijd;
   end klant;

process class verkoper;
   begin ref(klant) viseter;
         while true do
            begin
               while rij is niet leeg do
                  begin
                     viseter :- de volgende klant;
                     viseter.out;
                     reactivate viseter;
                     noteer gegevens;
                     hold(bedieningstijd);
                  end;
               passivate;
            end
   end haringverkoper;

ref(head) rij;
ref(verkoper) KLAAS;
maak de wachtrij;
maak de haringverkoper;
lees de modelparameters;
while time < simulatieperiode do
   begin
      trek aselecte getallen voor de aankomst- en bedieningstijd;
      activate new klant(tijd);
      hold(bedieningstijd);
   end;
hold(simulatieperiode + extra leeglooptijd);
report;
```

Hierna is het noodzakelijk zeker te weten dat deze modelbeschrijving goed is. Vooral in meer complexe programmamodellen is het aan te raden deze test door de computer te laten verwerken. In dat geval moeten nog niet ingevulde procedures, classes, gegenereerde objecten en statements worden vertegenwoordigd door dummies. Een dummy-statement is een qua syntax correcte opdracht, waarbij er echter geen acties worden uitgevoerd. Deze (detail-)acties worden pas later geformuleerd en in het programma ingevoerd. Belangrijk is dat het *gehele* hoofdprogramma aantoonbaar correct is. Bij het simpele voorbeeld als de haringkar, is het 'met de hand' doorlopen

van de programmabeschrijving doorgaans een voldoende test, hoewel men nooit volledige garantie krijgt.

De volgende modelbeschrijving geeft het complete SIMULA-programma.

```
Simulation
Begin comment de haringverkoper;
  process class haringverkoper;
    begin ref(klant) viseter;
      real som wachttijd;
      integer aantal klanten;
      while true do
      begin
        while not rij.empty do
        begin viseter :- rij.first;
          viseter.out;
          reactivate viseter;
          aantal klanten := aantal klanten + 1;
          som wachttijd := som wachttijd + viseter.wachttijd;
          hold(viseter.tijd);
        end;
        passivate;
      end;
    end haringverkoper;

  process class klant(tijd); real tijd;
     begin real tijd binnen,wachttijd;
       into(rij);
       tijd binnen := time;
       if klaas.idle
          then activate klaas
          else passivate;
       wachttijd := time - tijd binnen;
     end klant;

  procedure report;
    begin comment hierin worden de gevraagde gegevens overzichtelijk afgedrukt;
    end report;

  ref(head) rij;
  ref(haringverkoper) klaas;
  real aankomstgemiddelde, simulatietijd, aankomsttijd;
  integer A, B, start1, start2, tijd;

  rij :- new head;
  klaas :- new haringverkoper;
  A := inint; B:= inint;
  aankomstgemiddelde := inreal;
  simulatietijd := inreal;
  start1 := inint; start2 := inint;

  while time not greater simulatietijd do
  begin
    tijd := uniform(A, B, start1);
    aankomsttijd := negexp(1/aankomstgemiddelde, start2);
    activate new klant(tijd);
    hold(aankomsttijd);
  end;
  hold(simulatietijd + 10000);
  report;
End
```

V.8 AANWIJZINGEN VOOR HET PROGRAMMEREN

V.8.1 Software engineering

In de vorige paragrafen is aandacht besteed aan programmatuur en programmeertalen. Van zowel simulatie als van programmeren wordt wel gezegd dat het een kunst is. Op grond hiervan mag men mischien zeggen:

het programmeren van het simulatiemodel is een kunst met een grote K.

Toch zou het een slechte zaak zijn als de kwaliteit van een programma alleen bepaald zou worden door de kunstzinnigheid en creativiteit van een programmeur. Naast *kunst* is ook *kunde* een vereiste zoals we in het begin van § 3 al hebben opgemerkt. Een kunde die tot uitdrukking komt in een efficiënt gebruik van methoden en technieken om complexe programma's betrouwbaar te ontwikkelen. Voor het gebruik van deze methoden en technieken zijn evenwel beschikbaarheid en bekendheid eerste vereisten. Dat het aan beide nogal eens schort, komt tot uitdrukking in het volgende citaat*):

"De programmeertechnieken zijn bij deze groei (van programma's) in omvang en complexiteit sterk achtergebleven. Voor velen is programmeren nog steeds een **kunst** en is het nooit een **kunde** geworden. Een bijkomend probleem is dat veel programmeurs niet echt zijn opgeleid, maar het vak zelf aangeleerd hebben. Aan de organisatorische kant probeert men vaak een oplossing te vinden door het inzetten van steeds meer programmeurs, de zogenaamde *million monkey*-aanpak.

Als gevolg hiervan wordt programmatuur vaak te laat opgeleverd, doen de programma's niet wat de gebruiker ervan verwacht, zijn de programma's slecht aan nieuwe eisen aan te passen en ontdekt men na oplevering nog regelmatig fouten."

Gezien de in dit citaat genoemde problemen wordt de term softwarecrisis vaak gehoord. Aan dit onderwerp zijn verschillende artikelen en ook conferenties gewijd. Tijdens één van deze conferenties wordt de term **software engineering** gelanceerd voor het vakgebied dat zich bezighoudt met: **het ontwikkelen van methoden en technieken die toegepast kunnen worden bij de constructie van grote programma's.**

Waar het bij dit vakgebied om gaat, zo zegt Van Vliet**), zijn de volgende zaken:

---

*) Zie: J.C. van Vliet, Software Engineering, Leiden, 1984, pag. 1 en 2.

**) Zie: J.C. van Vliet, ibidem, pag. 4.

1. Het gaat bij software engineering om de constructie van grote programma's.
2. Het centrale punt is het overwinnen van de **complexiteit**.
3. Geregelde **samenwerking tussen mensen** is een integraal onderdeel van het programmeren-in-het-groot.
4. Programma's **evolueren**.

Omdat voor veel simulatieprogramma's de vier genoemde punten relevant zijn, is het gebruik van moderne werkwijzen uit de software engineering een noodzaak. Het is helaas onmogelijk in een boek als dit uitgebreid in te gaan op de methoden en technieken van software engineering. Daarvoor verwijzen we graag naar de speciale literatuur op dit gebied*). Wij zullen in de volgende paragraaf een viertal eisen formuleren waaraan een programma moet voldoen en enkele aanwijzingen geven om zo goed mogelijk aan deze eisen te voldoen.

V.8.2 Programma-eisen en aanwijzingen

We veronderstellen dat we beschikken over een goed ontwikkeld simulatiemodel, d.w.z. een model met een geheel doorzichtige structuur. Alle componenten zijn in de vorm van variabelen en parameters, de relaties zijn expliciet of impliciet geformuleerd. Er zijn geen duistere onbegrepen 'witte vlekken' in het model aanwezig. Als we over een helder, duidelijk model beschikken behoeft de vertaalslag naar het computerprogramma niet zo moeilijk te zijn. Het ontwerpen van zo'n programma is in wezen niets anders dan het vertalen van het symbolische model in een andere taal.

Belangrijk uitgangspunt is dan ook, dat het symbolische model volledig en correct moet zijn, alvorens de programmeerfase in te gaan.

Het programmeren zelf kan ook weer in enige fasen worden onderverdeeld. De eerste fasen zijn de probleemanalyse en het ontwerp. Bij de probleemanalyse worden naast de functies van het programma ook de eisen betreffende documentatie, testen en eventuele uitbreidbaarheid vastgelegd in de zogenaamde 'functionele specificaties'.

Bij het ontwerpvraagstuk zal men zeer gedisciplineerd te werk moeten gaan. Een simulatieprogramma is gewoonlijk van een niet geringe omvang. Gestructureerd programmeren houdt o.a. een zogenaamde top-down benadering in. Deze benadering betekent dat een globale opzet van het programma door een steeds verdergaande detaillering wordt gevolgd. Hierbij wordt tevens gekeken of het mogelijk is het programma te splitsen in redelijk onafhankelijke subsystemen of modulen die d.m.v. in- en uitvoer aan elkaar gekoppeld zijn. Deze 'hapklare brokken' kunnen dan afzonderlijk, eventueel door verschillende programmeurs worden uitgewerkt.

---

*) Zie bijvoorbeeld J.C. van Vliet, reeds aangehaald;
H.F. Ledgard e.a., Het groot Pascal Spreukenboek, Den Haag, 1982.

De top-down benadering vereenvoudigt ook de planning en de beheersing van het programmeringsproces. Tijd en mensen kunnen bij toepassingen nauwkeuriger worden geschat en toegewezen.

Efficiënte sturing van een programmeringsproject is mogelijk door tevens gebruik te maken van technieken als 'chief-programmer team operations' en 'structured walk throughs'. Ook worden hoge eisen gesteld aan standaardisering en documentatie.

Eisen waaraan elk programma, en dus ook een simulatie-programma moet voldoen, zijn de volgende.

1. Een programma moet correct zijn

Dat betekent dat het programma moet doen wat ervan verwacht wordt. Die verwachtingen zijn expliciet en duidelijk tijdens de probleemanalyse geformuleerd. Tevens is daarbij een testplan opgesteld.

De testinspanningen zullen afhankelijk zijn van het belang dat men aan de betrouwbaarheid van het programma hecht. Grofweg kan men fouten verdelen in syntactische, semantische en logische fouten.

Een computerprogramma dat 'werkt' is daarom nog geen goed programma. Maar 'werken' is toch wel het minste. Een programma wordt niet geaccepteerd door het vertaalprogramma (de compiler), als er nog taalfouten (syntaxfouten) inzitten. Wat het opvolgen van de grammaticaregels betreft, is de computer een strenge leermeester. Het is uiteraard een zaak van taalbeheersing om programma's te schrijven, die vanaf de eerste test foutloos zijn. In de praktijk zal het, vooral bij de eerste kennismaking met een taal, nogal eens voorkomen dat de compiler een 'waslijst' van syntaxfouten produceert. Toch moet men ernaar streven om programma's te schrijven, die reeds bij de eerste poging vrij zijn van syntaxfouten. De daartoe benodigde tijdsinvestering om de taal goed en secuur te beheersen, betaalt zich heel snel terug.

Veel moeilijker dan de syntaxfouten zijn de semantische fouten. Hierbij werkt het programma 'prima', maar het doet niet wat de bedoeling was.

Onder het begrip semantiek verstaan we de betekenis van de acties, zoals ze door de programma-opdrachten zijn weergegeven. Die betekenis moet men zelf door middel van de opdrachtbeschrijving aan de acties toekennen. Het vertaalprogramma kan alleen de grammatische correctheid van de opdrachten (syntaxis) beoordelen.

De controle op de logische fouten slaat terug op de probleemanalyse, eigenlijk zelfs op het simulatiemodel. We komen nu op het terrein van de validatie, waarop we in het volgende hoofdstuk dieper in zullen gaan.

Test- en verificatiemaatregelen dienen om fouten in het programma op te sporen. Helaas betekent het niet vinden van fouten geenszins dat het programma correct is. Hoe zorgvuldiger echter de fasen van probleemanalyse en ontwerp zijn doorlopen, des te groter zal de kans zijn op een foutloos programma.

Testtechnieken lopen van 'zorgvuldig lezen' (bij voorkeur door een ander) via 'walk-throughs' en 'correctheidsbewijzen' tot 'simulatie' (de werking van het programma wordt stap voor stap uitgevoerd met behulp van computerverwerking).

2. Een programma moet gemakkelijk te wijzigen zijn

Men noemt dit ook wel adaptief onderhoud.

Een simulatiemodel zal, wanneer het eenmaal geïmplementeerd is, dikwijls aangepast moeten worden aan de zich wijzigende omstandigheden van de omgeving (organisatie), waarin ze dienstbaar is. Het gaat dan vaak om nieuwe hardware of nieuwe systeemsoftware.

In veel gevallen kan na implementatie van het systeem nog blijken dat men wijzigingen in de systeemspecificaties wil aanbrengen. In de praktijk dient men er rekening mee te houden dat wijzigingen in het algemeen door andere mensen worden verzorgd dan de oorspronkelijke programmeurs. Dit leidt tot de volgende eis.

3. Het programma moet goed gedocumenteerd en leesbaar zijn

Er worden strenge eisen gesteld aan de documentatie, maar ook aan de leesbaarheid van het programma zelf.

Iedereen: management en gebruiker, projectleider en analist, ontwerper en programmeur zal zonder intensieve studie (snel) moeten kunnen begrijpen hoe het systeem gestructureerd is, hoe het werkt en wat ervan te verwachten is.

De relatie tussen de documentatie en het programma moet helder zijn. Het programma als zodanig is 'zelfdocumenterend'. De documentatie is geen aparte fase in het programmeringsproces, doch maakt onderdeel uit van elke fase. Het belang van documentatie, maar ook de noodzakelijke inspanningen voor het documenteren worden vaak onderschat.

4. Een programma moet betrekkelijk eenvoudig uitbreidbaar zijn

Soms worden aanvullende eisen gesteld aan de operationele en management-informatiesystemen waarvan simulatiemodellen deel uitmaken. Bij meer omvattende doelstellingen zal het nodig zijn om nieuwe gedeelten aan het model toe te voegen.

Nieuwe functies moeten snel, eenvoudig en zonder de structuur aan te tasten kunnen worden toegevoegd.

Om aan de genoemde eisen te kunnen voldoen is het noodzakelijk een aantal technieken te gebruiken. Een handzame en praktische leidraad wordt gegeven door Ledgard.

Wij volgen graag diens aanwijzingen.*)

...."De top-down benaderingswijze die hier wordt toegelicht is gebaseerd op de opvattingen van 'gestructureerd programmeren' van Dijkstra. Hoewel de methodiek geen wondermiddel is voor alle problemen bij het programmeren, biedt hij toch duidelijke richtlijnen voor een intelligente aanpak van een programmeringsprobleem.

Alvorens te coderen, moet iedere programmeur een volledige probleemstelling, een goed opgezet documentatiesysteem en een heldere ontwerpstrategie in handen hebben. De opmaak van de invoer, geldige en ongeldige velden, uitvoerbestanden, overzichten, programmaboodschappen en de ordening van alle verschillende situaties bij de invoer van gegevens tot en met de correcte uitvoer ervan moeten gedetailleerd worden beschreven. Bovendien moet het alles omvattende algoritme ook worden vastgelegd vóór de codering. Het is onzinnig om aan het coderen van een programma te beginnen zonder een dergelijke integrale benadering van het probleem.

Gegeven een deugdelijke probleembeschrijving en een totaal programmaontwerp, is de top-down benaderingswijze een methode om computerprogramma's in iedere programmeertaal te ontwikkelen. Kort samengevat kan deze methode als volgt worden gekenschetst:

1. Ontwerp in niveaus

De programmeur ontwerpt het programma in *niveaus*, waarbij een niveau bestaat uit één of meer modules. Een module is altijd 'compleet', hoewel deze kan verwijzen naar ongeschreven modules. Het eerste niveau is een volledig 'hoofdprogramma'. De lagere niveaus verfijnen of ontwikkelen de ongeschreven modules van de hogere niveaus. Met andere woorden: de modules van een opvolgend niveau bestaan uit de submodules waarnaar werd verwezen in het eerdere niveau. De programmeur kan bij het ontwerpen van een niveau verscheidene niveaus vooruit kijken om zo tot de beste structuur te komen.

2. Onafhankelijkheid van de taal in het begin

De programmeur gebruikt in het begin uitdrukkingen die van belang zijn voor de probleemoplossing, zelfs al kunnen deze uitdrukkingen niet direct in code worden overgebracht. Van instructies die onafhankelijk zijn van machine en taal werkt de programmeur toe naar een machinegerichte versie in een programmeertaal.

3. Uitstel van details tot lagere niveaus

De programmeur concentreert zich op de kritische hoofdpunten van de eerste niveaus en stelt details (bijvoorbeeld de keuze van specifieke algoritmen of de weergave van tussenresultaten) uit tot lagere niveaus.

---

*) Zie H.F. Ledgard e.a., ibidem, pag. 93-95.

4. Formalisering van ieder niveau

Alvorens met een lager niveau verder te gaan, verzekert de programmeur zich ervan, dat het programma in zijn huidige ontwikkelingsfase 'formele' instructies bevat. In de meeste gevallen betekent dit een programma dat ongeschreven submodules inclusief alle nodige parameters oproept. Deze stap biedt de zekerheid dat verdere programmasecties onafhankelijk zullen worden ontwikkeld, zonder dat later specificaties en interfaces (raakvlakken) tussen modules gewijzigd moeten worden.

5. Verificatie van ieder niveau

Na de modules van een nieuw niveau geproduceerd te hebben, verifieert de programmeur het formele programma tot zover als het nu ontwikkeld is. Dit biedt de zekerheid, dat fouten inherent aan het actuele niveau van ontwikkeling opgespoord zullen worden op hun eigen niveau.

6. Opeenvolgende verfijningen

Ieder niveau van het programma wordt verfijnd, geformaliseerd en geverifieerd in opeenvolgende versies, totdat de programmeur het volledige programma heeft verkregen dat gemakkelijk in Pascal of een andere taal kan worden omgezet.

Nog een aantal opmerkingen over de top-down benaderingswijze. Ten eerste wordt verondersteld, dat het gehele probleem en de integrale oplossing ervan zijn doordacht en begrepen. Het is zinloos om al te gaan programmeren voordat het probleem geheel is begrepen en een integrale strategie is ontwikkeld. Een dergelijk begrip stelt de programmeur in staat het programma te schrijven zonder het zicht te verliezen over het hoofddoel.

Ten tweede is in de eerste versies van de hoogste niveaus de benaderingswijze onafhankelijk van machine en taal en wordt de programmeur niet beperkt door de details van een programmeertaal. Hij of zij schrijft de hoogste niveaus uit en gebruikt daarbij een notatie die het probleem op een betekenisvolle wijze oplost. Bij de keuze van zo'n specifieke notatie wordt geen enkele concessie gedaan aan de machine. Op ieder niveau vertegenwoordigen de geproduceerde instructies nog steeds een volledig programma in zekere zin. Alles wat ontbreekt is de machine die in staat is de instructies uit te voeren.

Ten derde moet op ieder ontwerpniveau in volgende versies de informele notatie worden geformaliseerd tot een hypothetische, maar expliciet gespecificeerde procedure. Specificatie houdt een volledige opsomming in van alle in- en uitvoerparameters.

Ten vierde moet de programmeur op ieder niveau het programma verifieren in zijn actuele vorm, zodat verdere verfijningen volledig correct zullen zijn met betrekking tot eerder ontworpen niveaus. Dit biedt de zekerheid dat tekortkomingen zullen worden ontdekt in hun eigen context."

HOOFDSTUK VI

Validatie

VI.1 INLEIDING

In de voorgaande hoofdstukken is het simulatie-onderzoek faseringsgewijs behandeld. Voordat men een nieuwe fase begint, dient het voorgaande werk te worden getoetst. De beschouwingen over de juistheid en geldigheid van het werk vallen onder het begrip validatie. In de globale driedeling: inductie-deductie-validatie wordt gesuggereerd dat validatie één hoofdfase is, welke men aan het eind van de rit dient af te handelen. Het blijkt echter, zie bijvoorbeeld figuur III.2b, dat men voortdurend moet valideren en terugkoppelen om na te gaan of men op het goede spoor zit. We hebben gezien, dat zich tijdens de modelontwikkeling diverse valkuilen kunnen voordoen die aanleiding kunnen geven tot het trekken van foutieve conclusies:

1. fouten in het ontwerp (beeldsysteem en probleemspecificatie)
2. fouten in de programmering
3. fouten in de gebruikte gegevens (waarnemingen)
4. fouten in het gebruik van het model en de wijze van hanteren
5. fouten in de interpretatie van de modeluitkomsten.

Deze foutenbronnen moeten steeds in de gaten worden gehouden. Dientengevolge is het nodig om tijdens het gehele simulatieproject (de modelcyclus) voortdurend te evalueren.

Deze gedachtengang leidt tot de '*multistage verification*', waarbij in drie stadia het evaluatieproces als een onverbrekelijke combinatie met het ontwerpproces wordt doorlopen. In hoeverre de mogelijkheden tot validatie echter toereikend zijn, is zeer afhankelijk van de concrete onderzoeksituatie waarin men zich bevindt. Men zal vaak verder gaan, terwijl er nog een behoorlijke mate van onzekerheid aanwezig is in het tot dan toe verkregen model.

Pas wanneer het model werkt, d.w.z. in staat is om 'response' te leveren, kunnen aan de hand van de resultaten definitieve uitspraken omtrent de geldigheid worden gedaan. Dit resultaten-onderzoek is tevens de derde hoofdplaats van validatie in de simulatiecyclus.

In § 2 wordt een algemene beschouwing gewijd aan de geldigheid van een model. Daarnaast wordt aan de hand van een schets van de modelcyclus aangegeven welke de plaats en de inhoud zijn van de diverse validatievormen. De inhoud van de drie validatievormen wordt vervolgens in § 3 uitgediept. Tevens wordt daar aangegeven welke toetsen kunnen worden gebruikt.

§ 4 gaat over een meer specifieke validatiemogelijkheid: gevoeligheidsanalyse. Deze vorm is van belang zijn voor de uiteindelijke acceptatie van het model door de opdrachtgever/gebruiker.

De slotparagraaf heeft een meer filosofisch karakter. In deze paragraaf wordt het vraagstuk van de validatie vanuit een historisch wetenschapsfilosofische invalshoek belicht. Stromingen als het rationalisme, empirisme en pragmatisme worden kort besproken op grond van hun belangrijkste kenmerken. Elk van deze drie stromingen geeft een vertrekpunt van waaruit men het onderwerp validatie kan bespreken. Aan de hand van de door ons onderscheiden drie validatiefasen wordt een synthese van de stromingen besproken.

VI.2 DE GELDIGHEID VAN EEN (SIMULATIE)MODEL

We zouden van een simulatie-onderzoek graag willen zeggen dat het 'waar' is, geldig en foutloos. We moeten op ons model kunnen vertrouwen en er voor instaan dat de onderzoekresultaten juist zijn. Deze verschillende, sterke gerelateerde begrippen zijn niet zo gemakkelijk eenduidig te definiëren.

Omschrijvingen van de begrippen geldigheid en betrouwbaarheid luiden:*)

"Onder de *geldigheid* van een onderzoek wordt de mate verstaan waarin de theoretische begrippen, opgevat als theorieën, werkelijkheidswaarde en verklaringswaarde hebben ten opzichte van verschijnselen in de werkelijkheid en de mate waarin de operationaliseringen in hun betekenis met de theoretische begrippen overeenstemmen".

"Onder de *betrouwbaarheid* van onderzoek wordt de mate verstaan waarin de onderzoekresultaten onafhankelijk zijn van de technische uitvoering van een onderzoek".

Betrouwbaarheid kan men beter kwantificeren dan geldigheid: Zo kan men hierover afspraken maken in statistische zin, bijvoorbeeld met betrouwbaarheidsintervallen.

Daar simulatie het bouwen van een model inhoudt, gaat het erom dat de eisen, die men aan het model stelt, afgeleid moeten zijn uit dezelfde doelstellingen, die geleid hebben tot het model. Zo zullen de betrouwbaarheid en de geldigheid van het model moeten worden uitgedrukt in termen die op de doelstellingen betrekking hebben. Deze relatie wordt vaak verwaarloosd of tenminste niet expliciet gemaakt.

In algemene zin is de hoofddoelstelling van simulatie een model te bouwen dat dezelfde gedragseigenschappen heeft als het systeem dat we wensen te onderzoeken. De problemen uit de werkelijkheid die opgelost moeten worden, dienen dezelfde problemen te zijn, waarover m.b.v. het model uitspraken worden gedaan. Pas als dat geldt, kan men van een valide model spreken, m.a.w. met validatie wordt de relatie: gedrag van het model - gedrag van het reële systeem getoetst.

Om te kunnen spreken van een valide model moet men een zeker acceptatieniveau van vertrouwen bereiken zodat men kan aannemen, dat conclusies die men wenst te trekken uit de uitkomsten van het model correct en toepasbaar zijn op de realiteit.

De term 'acceptatieniveau' duidt erop dat een geldigheid van 100% een problematische zaak is. Het duidt ook op een zekere

---

*) R.W. Boesjes-Hommes, De geldige operationalisering van begrippen, Meppel, 1970.

gradatie: een model zal zich gewoonlijk bewegen tussen volledige geldigheid en volledige ongeldigheid. In hoeverre, als een model voldoet, van geldigheid sprake is, zal overigens nooit met zekerheid te zeggen zijn. Het is in dit verband de moeite waard om eens de gedachten van de bekende filosoof Popper te lezen. Hij vergelijkt de realiteit met een gebergte waarvan de toppen ver weg zijn en onbekend. Een onderzoeker is als een bergbeklimmer: hij klautert voortdurend omhoog, maar komt steeds tot de conclusie dat de hoogste top nog niet is bereikt. Het gebergte blijft zekere geheimen houden. Datzelfde geldt voor het reële systeem dat we onderzoeken. We bestuderen de werkelijkheid aan de hand van modellen, maar of zij de werkelijkheid in al haar klaarheid aan de onderzoeker kunnen tonen valt nooit definitief te zeggen. Zolang een model voldoet, kunnen we hooguit hopen dat we met een geldige afbeelding te doen hebben. Het model is dan niet gefalsificeerd, niet weerlegd door de feiten.

Het verhogen van de geldigheidsgraad van een model zal zonder meer een verhoging van de kosten met zich brengen. Hoe meer een model wordt uitgebreid des te uitgebreider zal het onderzoek moeten zijn. Het aantal componenten en het aantal relaties nemen toe. Wellicht zal ook de waarde van het model toenemen, maar na een bepaald punt zal elke toename met een steeds grotere toename van de kosten gepaard gaan.

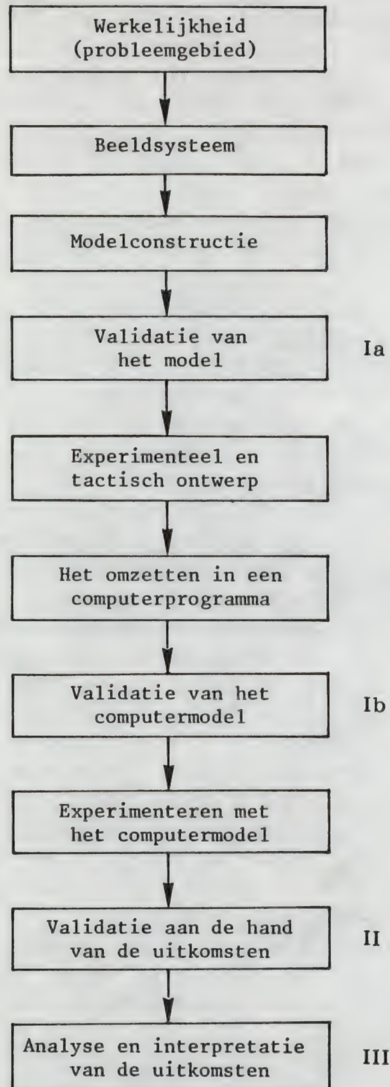
Het voorgaande betekent dat men zich, gegeven de doelstelling van het onderzoek en gegeven het probleem goed moet realiseren welke graad van geldigheid, bijvoorbeeld uitgedrukt in termen van nauwkeurigheid en betrouwbaarheid gewenst is.

Stel bijvoorbeeld dat er een simulatie-onderzoek is, waarbij beslissingsregels voor het financieel management van een bankbedrijf worden geëvalueerd. Bij het betalingsverkeer mogen geen fouten worden gemaakt. Aan het model, dat hiervan een afbeelding is, worden hoge eigen gesteld t.a.v. de geldigheid.

Het management moet het volste vertrouwen hebben, zodat ze als het ware 'blind' kan varen op de resultaten van het model. Is daarentegen een model gebouwd, waarmee strategieën voor een complex wachtrijprobleem worden vergeleken, bijvoorbeeld bij een job-shop-simulatie, dan zal men t.a.v. de absolute uitkomsten geen bovenmatige nauwkeurigheid verlangen.

Er zijn vele factoren die de moeilijkheidsgraad van het valideren beïnvloeden. We komen daar later op terug, doch we willen nu reeds noemen, dat de minst beheersbare factoren de factoren met stochastische eigenschappen zijn. Altijd zijn in het geval van stochastische systemen 'random' 'fouten' aanwezig, waarvan men de invloed probeert te verkleinen door het nemen van steekproeven van grotere omvang of door de experimenten een groot aantal malen te herhalen. Maar omdat de aangenomen en eventueel getoetste verdelingsfuncties, die we plegen te gebruiken, benaderingen zijn, blijft ondanks geavanceerde statistische analyses een impliciete onzekerheid in de uitkomsten van het model ingebouwd. Overigens geldt dit ook voor de werkelijkheid.

Op verschillende plaatsen in het simulatie-onderzoek wordt, zoals reeds eerder is opgemerkt aandacht geschonken aan de validatie. Wanneer we wederom een schets maken van de modelcyclus dan kunnen we het onderscheid aangeven t.a.v. de plaats en de inhoud van de verschillende validatievormen.



Figuur VI.2a De modelcyclus*)

*) Vergelijk dit schema nog eens met de figuren III.2c en IV.2a.

In dit schema is op vier plaatsen expliciet aangegeven dat er een onderzoek naar de juistheid van de produkten uit de voorafgaande werkfasen moet worden gedaan.

We onderscheiden daarin een drietal categorieën. We beschouwen eerst de juistheid van het model (Ia en Ib), vervolgens wordt na het uitvoeren van de experimenten gekeken of het model ook geldig is (II) en tenslotte zullen de interpretatiemogelijkheden (III) de uiteindelijke waarde van het onderzoek bepalen.

I *Verificatie en Consistentie*

Bij de constructie van het model wordt een verzameling hypothesen gedefinieerd, gebaseerd op alle mogelijke informatie, zoals waarneming, vroeger onderzoek, relevante theorie en intuïtie. De consistentie en de plausibiliteit zullen bij de verificatie aan de orde moeten komen.
De subjectiviteit van de modelconstructie wordt hierbij niet doorbroken. M.a.w. het gaat hier om het onderzoek of het model overeenkomt met de interpretatie van de onderzoeker van het reële systeem.

II *Validatie*

De toets van de overeenkomst tussen het gedrag van het model en het reële systeem. Het gaat nu om de acceptatie van het model. De vraag naar de voorspellende waarde van het model is hier de belangrijkste.
Hiervoor zal van statistische procedures gebruik gemaakt moeten worden.

III *Resultatenanalyse*

Het gehele onderzoek wordt gevalideerd. Het gaat nu om de acceptatie van het beeldsysteem. Deze validatievorm dient dus betrekking te hebben op de probleemspecificatie als de belangrijkste basis voor het beeldsysteem. De uitkomsten van het simulatiemodel worden geïnterpreteerd en liggen vaak ten grondslag aan het beslissingsproces van de gebruiker.

Het is niet altijd even eenvoudig om de verschillen in de validatiecategorieën scherp te zien. Zo lijken II en III samen te vallen, omdat beide validatievormen aan de hand van de uitkomsten worden uitgevoerd, maar dit is niet het geval.

Validatie II is als het testen van het model onafscheidelijk verbonden met de bouw van het model: ze vormen samen het wetenschappelijk onderzoek. Pas na de acceptatie van het model worden de resultaten geïnterpreteerd en mag men conclusies proberen te trekken. Overigens zij hier opgemerkt, dat ook de validatie I (de verificatie) pas volledig mogelijk wordt nadat experimenten met het model voor toetsbare uitkomsten hebben gezorgd.

Uit het bovenstaande betoog volgt nogmaals de reeds enkele malen geplaatste opmerking, dat men een aantal fasen van het

simulatieproject meerdere malen zal moeten doorlopen, afhankelijk van wat de betreffende validatievorm oplevert. Men kan dus inderdaad van een cyclus spreken.

We zullen in de volgende paragraaf aan elk van de validatievormen een korte beschouwing wijden.

VI.3 DRIEMAAL VALIDATIE

VI.3.1 Modelconsistentie en verificatie

Bij de ontwikkeling van het (simulatie)model worden uitspraken gedaan over de in het model op te nemen variabelen, over de relaties en de parameters. Deze uitspraken worden zoveel als mogelijk gedaan naar aanleiding van een grondige analyse van de werkelijkheid en worden geformuleerd als hypothesen.

Met name twee begrippen spelen als het gaat om de validatie in deze fase een rol, namelijk *plausibiliteit* en *consistentie*.

Met consistentie wordt de mate van samenhangendheid bedoeld. In de logica heet een theorie consistent als er geen tegenspraak of contradictie in afgeleid kan worden. Als dus uit een model tegenstrijdige uitspraken zijn af te leiden, is het model inconsistent. Het bewijs van consistentie is niet zo eenvoudig. Veelal zal, zeker in de aanvangsfase van de modelvorming gebruik worden gemaakt van intuïtie. Vooral als het gaat om het opstellen van relaties is er sprake van een ingewikkeld vraagstuk. Alvorens daarom over consistentie te spreken, is het verstandig eerst te streven naar plausibiliteit van het model.

Om na te gaan of de veronderstellingen van de relaties in het model plausibel zijn, kent men een aantal toetsen. Bekend is o.a. de experttoets. Bij toepassing hiervan wordt een aantal deskundigen naar hun mening gevraagd over het gedrag van het onderzochte systeem bij een gegeven invoer. Wanneer deze mening vergeleken wordt met de uitkomsten van het model en de afwijkingen zijn groot, dan kan dit een reden zijn om het model te verwerpen.

Een tweede toets vergelijkt het gedrag van het systeem in de tijd met dat van het model. Als de fluctuaties groter zijn dan men aanvaardbaar acht, dan kunnen de hypothesen omtrent de relaties onjuist zijn gespecificeerd of het is mogelijk dat variabelen of relaties niet in het model zijn opgenomen. Een dergelijke toets wordt een stabiliteitsproef genoemd. Het systeem in onderzoek kan zelf ook tijdsvariant zijn, d.w.z. dat de structuur of de variabelen veranderen in het verloop van de tijd. Vooral bij organisatie-onderzoek (denk in dit geval aan systeemanalyse t.b.v. automatisering) zal dit kunnen voorkomen.

In de genoemde gevallen worden de simulatie-uitkomsten vergeleken met resultaten van het reële systeem (historical output). Hiervoor is het noodzakelijk dat van minstens één variant van het simulatiemodel het systeem in werkelijkheid bestaat.

Deze validatie Ia zal in het geval dat met computers gewerkt wordt *voorafgegaan* moeten worden door de validatie Ib, welke 'slechts' een technische validatie is t.a.v. het computerprogramma. Ze bestaat uit een syntactische analyse, geheel door de computer of interpretator verzorgd, een semantische analyse, waarvoor de computer slechts ten dele ondersteuning kan bieden (logische fouten in het ontwerp die naar 'runtime errors' leiden) en de pragmatische

analyse, waarbij de onderzoeker zelf moet zien of het computerprogramma wel een 1:1 afbeelding is van zijn ontworpen model.

VI.3.2 Validatie van het model

De validatievorm II betreft de *overeenkomst* van het model met het reële systeem.

Hierbij is het wenselijk om eerst stil te staan bij de gedachte of het noodzakelijk is dat het model werkelijk isomorf is met het onderzochte systeem.*) Een model kan ook heel simpel zijn en toch voldoen aan de eisen die wij eraan stellen.

Er bestaan voorbeelden van heel eenvoudige voorraadregels, die uitstekend werken voor een complex voorraadsysteem. Er bestaan betrekkelijk eenvoudige macro-economische modellen met een aanvaardbare verklarende kracht.

De vraag: 'hoe geldig is het model' wordt in feite bepaald door de toepassingsmogelijkheden. Het doel van het model staat dan voorop.

De validatietoets waarvan hier sprake is, hanteert het probleem van de 'gewantrouwde waarneming', dat wil zeggen we nemen alles voor juist aan totdat het tegendeel blijkt. Het beroemde voorbeeld is dat van Einsteins relativiteitstheorie, waarmee de 'eeuwig geldende' natuurwetten van Newton als onjuist bewezen werden.

Hoe dan ook, het zijn waarnemingen op grond waarvan het model ontwikkeld wordt en het zijn eveneens waarnemingen, waarmee naderhand de uitkomsten van het model vergeleken worden, zodanig dat men uitspraken kan doen omtrent de geldigheid van het model. Men moet er hierbij wel op letten dat de waarnemingen, die gebruikt worden in beide genoemde fasen twee disjuncte groepen vormen.

Ideaal zou zijn dat men het model opbouwt op een waarnemingsreeks uit het verleden en valideert m.b.v. toekomstige waarnemingen. Men valideert als het ware de voorspellende waarde van het model.

Het grote gevaar hiervan is natuurlijk dat het vaak lang (soms te lang) kan duren, voordat men met voldoende betrouwbaarheid uitspraken kan doen over de validiteit. Daarom zal deze 'prospectieve toets' vaak alleen maar toepassing vinden in de continue evaluatie van de modelprestaties, wanneer het model reeds geaccepteerd en geïmplementeerd is.

Overigens willen we met nadruk wijzen op de noodzakelijkheid van een continue evaluatie, daar het reële systeem, dat met behulp van het model nagebootst wordt niet onveranderlijk is.

---

*) We verstaan hieronder dat het model een één-éénduidige afbeelding van het systeem is. Bij elk object van het systeem hoort een object van het model en bij elke relatie van het systeem hoort een relatie van het model. Dit behoeft lang niet altijd het geval te zijn.

Terugkomend op het validatievraagstuk: er zal gewoonlijk met beschikbare 'historische gegevens' moeten worden gewerkt. In dit geval noemen we de toets retrospectief (postdictie). We maken dan gebruik van een groep historische waarnemingsuitkomsten om de variabelen, de relaties en de parameters van een inhoud te voorzien en van een *andere* groep historische waarnemingsuitkomsten om het model te toetsen.

Zou men deze splitsing niet maken, dan heeft de toets geen waarde!

N.B. Vergeet overigens óók niet om het waarnemingsmateriaal te toetsen.

Er bestaat een grote verzameling statistische en grafische toetsen om simulatiemodellen te valideren. Om een weloverwogen keuze uit de beschikbare hoeveelheid te kunnen doen, is de vraag wat men wil toetsen van belang.

In wezen is de validatievorm II een test van de interne structuur. Bij de bouw van het model zijn hypothesen gesteld, die getoetst moeten worden. Men moet dientengevolge bedacht zijn op de mogelijke tekortkomingen waaraan het model kan lijden.

Aan Ackoff en Sasieni*) hebben we de volgende punten ontleend.

1. Het model kan irrelevante variabelen bevatten; deze beschrijven het gedrag dus niet of niet juist.
2. We zijn vergeten om een of meer relevante exogene variabelen, die het gedrag wel beïnvloeden, in te bouwen.
3. Eén of meer van de relevante variabelen worden onjuist geëvalueerd of gerepresenteerd.
4. Eén of meer van de functionele relaties tussen de exogene en endogene variabelen zijn onjuist geformuleerd.
5. De schattingen van de parameters zijn onjuist.

Ten behoeve van het toetsen van hypothesen noemen we een aantal statistische technieken.

1. Het toetsen van schattingen van parameters, waarbij een bepaalde waarschijnlijkheidsverdeling kan worden aangenomen. Voorbeelden hiervan zijn de u-toets, de t-toets en de F-toets.
2. Het toetsen van schattingen van parameters zonder dat een verdeling kan worden aangenomen: de parametervrije toetsen, zoals de Mann-Whitneytoets, de tekentoets en de toets van Wilcoxon.

---

*) R.L. Ackoff en M.W. Sasieni, Fundamentals of Operations Research, New York, 1968.

3. Het toetsen van verdelingen waaruit steekproefwaarnemingen worden getrokken. Men spreekt in dit verband van de 'Goodness of fit'. Voorbeelden hiervan zijn de chi-kwadraat en de Kolmogorov-Smirnov test.

4. Het toetsen van de samenhang en van de relaties: correlatie- en regressie-analyse.

VI.3.3 Resultatenanalyse

Nadat getest is of het model goed voldoet als afbeelding van het reële systeem zullen aan de hand van experimenten de prestaties van het model worden getest. Deze prestaties zullen worden gemeten in termen van betrouwbaarheid, nauwkeurigheid en volledigheid. Men moet nu in staat zijn om conclusies te trekken en beslissingsregels af te leiden.

Belangrijke grootheden aan de hand waarvan de resultaten beoordeeld zullen worden, zijn het gemiddelde en de variantie. De manier om de nauwkeurigheid van de resultaten te vergroten is een verlaging van de variantie. Zoals reeds eerder is opgemerkt, is één van de methoden om dit te bereiken een vergroting van de steekproefomvang. Dit is evenwel duur, zodat naar andere methoden is gezocht om de variantie te reduceren. In de statistische appendix bespreken we twee andere technieken voor variantiereductie.

Ook willen we hier nogmaals verwijzen naar zaken als 'run'-lengte en startwaarden die van belang zijn voor validatie van de resultaten.

Doorgaans verkrijgt men uit het simulatie-experiment grote hoeveelheden 'waarnemingsmateriaal'. Het is de bedoeling om met behulp van statistische analyse dit materiaal te ordenen en de conclusies in de vorm van geaggregeerde informatie te presenteren. De meest bekende analysemethoden zijn:

- Regressie-analyse (inclusief discriminant- en factoranalyse);
- Variantie-analyse;
- Spectraalanalyse.

Spectraalanalyse heeft als nadeel wiskundig vrij moeilijk te zijn. Men kan evenwel een benadering toepassen om tijdreeksen te vergelijken, zoals ze zijn geproduceerd door het werkelijke systeem en door het model. De vergelijking kan gebeuren op de volgende punten:

1. het aantal omslagpunten

2. de momenten in de tijd van de omslagpunten

3. de richting van de reeks in de omslagpunten

4. de amplitude van de bewegingen voor corresponderende gedeelten uit de reeks

5. de fase van de bewegingen
6. de gemiddelde waarde van de variabelen
7. de verdelingsfuncties van die variabelen
8. de overeenkomst in plaats en tijd van de omslagpunten van de variabelen.

Een aantal van de genoemde punten is met parametervrije toetsen te onderzoeken.

Omdat elk van de analysemethoden tijdrovend is, zal men bij de analyse van het cijfermateriaal zoveel mogelijk gebruik moeten maken van statistische pakketten, zoals SPSS en WESP. Ook voor de presentatie van de uitvoer, bijvoorbeeld in de vorm van tabellen en van grafieken kan van speciale programmapakketten gebruik worden gemaakt.

Dat men in dit laatste stadium van validatie vooral de beeldconstructie evalueert, wordt door Bosman*) verduidelijkt:

> "In een situatie, waarin simulatie als zelfstandige onderzoekmethode wordt gebruikt, zal ook de beeldconstructie, vooral in samenhang met de simulatie in de inductieve fase, in de validatiebeschouwingen moeten worden betrokken.
> Het proces van valideren van beeldconstructies kan niet worden beschouwd als een direct toepasbare procedure. Het gehele proces van wetenschappelijk denken, bijvoorbeeld in de vorm van het regelmatig doorlopen van de modelcyclus, komt uiteindelijk neer op een proces, dat inhoud moet geven aan het beeld.
> De stelling, dat het ene model een betere representatie van de werkelijkheid is dan het andere, is daarom aanvechtbaar. Wel kan men concluderen, dat het ene systeem, gegeven dezelfde probleemdefinitie en vergelijkbare validatieprocedures, betere resultaten oplevert dan het andere."

---

*) A. Bosman, Een metatheorie over het gedrag van organisaties, Leiden, 1977.

VI.4 GEVOELIGHEIDSANALYSE

Gevoeligheidsanalyse is een niet te onderschatten hulpmiddel bij het besluitvormingsproces.

Met name bij het planningsproces zal gevoeligheidsanalyse (Engels: sensitivity analysis) onontbeerlijk zijn. Gewoonlijk verstaat men hieronder het systematisch variëren van waarden van variabelen en parameters, om op deze wijze de invloed ervan op het gedrag van het systeem te bestuderen.

Het systeem zal op verschillende variabelen, of op verschillende combinaties van variabelen anders reageren. Voor sommige factoren is het systeem heel gevoelig, d.w.z. dat kleine wijzigingen in een factor grote gevolgen voor de responsevariabelen hebben.

Uiteraard kunnen experimenten die zo'n type analyse ten doel hebben, het vertrouwen in het model aanzienlijk ondersteunen. Immers, juist met gevoeligheidsanalyse kunnen we aantonen of een vooronderstelling al dan niet juist is.

In de praktijk blijkt dat het merendeel van de exogene variabelen vaak weinig of geen invloed heeft op het gedrag van het model. Het is begrijpelijk dat het voor een goede besluitvorming wenselijk is nauwkeurig te weten, welke variabelen van belang zijn: de zogenaamde relevante exogene variabelen.

Dit is ook de reden waarom gevoeligheidsanalyse in het bovengenoemde planningsproces thuishoort.

Gevoeligheidsanalyse is met name juist in simulatie zo goed toepasbaar, omdat de analist het model onder volstrekte controle heeft. Anders dan in het reële systeem kan men van elke willekeurige factor de invloed op de endogene variabelen van het model meten.

Wanneer we de toepasbaarheid van gevoeligheidsanalyse op een meer systematische wijze beschouwen, dan kunnen we vier interessante aspecten onderscheiden.

1. De gevoeligheidsanalyse verschaft ons informatie over de mate van gevoeligheid van de resultaten van het model voor de gebruikte waarden van de exogene variabelen. Omdat we vaak weinig kennis bezitten omtrent de gegevens, waarop we het model hebben gebaseerd, moeten we die kennis kunnen aanscherpen.
   Als blijkt dat variatie in variabelen of parameters weinig effect op het modelgedrag opleveren, dan behoeven we ons omtrent de nauwkeurigheid van deze verzameling variabelen niet al te druk te maken. Als het model daarentegen zeer gevoelig blijkt voor een bepaalde variabele of parameter, dan zullen we geneigd zijn tijd en moeite te investeren om deze relevante grootheid zo nauwkeurig en zo realistisch mogelijk in kaart te brengen.

2. Omdat we met gevoeligheidsanalyse de invoerwaarde over een groot waardenbereik kunnen variëren, is het mogelijk hierdoor

informatie te krijgen over de invloed van een veranderende omgeving op de responsevariabelen.
Wanneer we weten dat het systeem tamelijk ongevoelig is voor storingen dan sterkt dat het vertrouwen om de besluitvorming op de resultaten van het model te baseren. Komen er bepaalde variabelen voor, die wel op dramatische wijze ons model beïnvloeden, dan zullen we bij de implementatie hier goed op moeten letten en zo mogelijk corrigerende maatregelen, dan wel een selectief meetsysteem moeten inbouwen.

3. De gevoeligheidsanalyse verschaft ons belangrijke aanwijzingen voor mogelijke wijzigingen, die later in het model aangebracht kunnen worden. Immers, als het model een werkelijk geschikt hulpmiddel voor de besluitvorming zal blijken te zijn, is er een continu proces van aanpassing en wijziging nodig. Probleemspecificaties zijn namelijk niet constant, doch hebben de neiging steeds 'weg te lopen'.
Overigens stelt het bovenstaande wel hoge eisen aan de structuur van het model; flexibiliteit en modulariteit zijn van primair belang.
De gevoeligheidsanalyse kan dus ook aanwijzingen geven omtrent de manier waarop het model gestructureerd moet worden, opdat het bestand is tegen voortdurende aanpassing.

4. Wanneer de gebruiker/opdrachtgever geconfronteerd wordt met de resultaten van de gevoeligheidsanalyse kan dat een enorm positief psychologisch effect hebben, aangezien hem dan exogene variabelen met betrekking tot de uitvoervariabelen duidelijk wordt. De acceptatie wordt daardoor aanzienlijk versneld, omdat zodoende een zekere geruststelling aangaande de betrouwbaarheid van het model wordt gegeven.

VI.5 HISTORISCHE UITGANGSPUNTEN; een SYNTHESE

Eén van de grondslagen van wetenschappelijk onderzoek is de eis van objectiviteit. Ook simulatie-onderzoek moet aan deze eis voldoen.

Daar echter simuleren noodzakelijkerwijs een model behoeft, is het ontwerpen van het model een fase die in het onderzoek onontkoombaar is. Hier ligt de bron van het conflict: het ontwerpen van een model is een subjectieve bezigheid. Bij het ontwerpen en valideren van simulatiemodellen is er een conflict tussen enerzijds de noodzaak om objectief te zijn en anderzijds de noodzaak om een constructief en intelligent gebruik te maken van onze eigen subjectieve eigenschappen. Met dit laatste worden eigenschappen bedoeld als ervaring, opleiding, intuïtie, opvattingen, intelligent giswerk en de kunst van het waarnemen van de reële systemen.

Objectief dienen we de subjectieve opvattingen, indrukken en vooroordelen terzijde te schuiven en slechts datgene te beschouwen, wat zich experimenteel openbaart.

Of, volgens C.R. Blyth*): "Juist het grote dilemma van wetenschappelijk onderzoek is, dat de onderzoeker objectief moet zijn, maar dat hij alleen vordering maakt in zijn onderzoek door zijn subjectieve inzichten te volgen".

Het belang van validatie is in feite, dat zo de subjectieve modellen worden geobjectiveerd. Het ontwerp en de validatie van een model betekenen een voortdurende heen- en weergang tussen een subjectieve en een objectieve fase.

Het doel van validatie is te bepalen of het model objectieve toetsen kan weerstaan, zonder dat onze (voor-)oordelen hierbij een rol spelen. Hiermee is een andere verklaring gegeven voor de reeds enige malen geponeerde stelling, dat de modelcyclus bestaat uit een afwisseling der fasen waarin subjectieve en objectieve elementen een rol spelen.

Hoe moeten we nu aan het werk gaan? Wat is nu verantwoord? In de wetenschapsfilosofie kunnen voor de beantwoording van deze vragen twee hoofdstromingen worden gevonden, namelijk het rationalisme en het empirisme.

De tegenstelling tussen deze twee richtingen spitst zich toe op de vraag, of het geoorloofd is bij de modelvorming alleen af te gaan op het verstand dan wel alleen op de ervaring. Beide richtingen kunnen bogen op beroemde vertegenwoordigers. Zo zijn Descartes en Spinoza grondleggers van het rationalisme en Locke en Mill vertegenwoordigers van het empirisme. Om enig idee te geven van de denkwijze van beide richtingen geven we een korte beschrijving.

---

*) C.R. Blyth, in: The American Statistician, "Subjective vs. objective methods in statistics", Vol. 26, nr. 3, juni 1973.

De rationalist baseert zich in hoge mate op wiskunde en logica. Hij ontwikkelt een mathematisch model, waarin de hypothese de vorm krijgt van een in wiskundige vorm gegoten beschrijving van de waarnemingen. Het rationalistische model bestaat uit een systeem van logische afleidingen, die voortkomen uit een verzameling premissen.

Een hedendaags voorbeeld van rationalisme is de theorie van Forrester*). Zijn urbane modellen zijn gebaseerd op rationalistische uitgangspunten. Zelf zegt hij ervan, dat het veel ernstiger is om een relatie weg te laten, waarvan men aanneemt dat deze belangrijk is dan wel haar in het model op te nemen met acceptatie van een lage nauwkeurigheid. Uiteraard brengt deze aanpak nadelen met zich mee; je kunt tenslotte wel eens ongelijk hebben.

Het andere uiterste, de empirische benadering, weigert enige vooronderstelling of aanname te doen, die niet door een experiment of door middel van analyse van empirische gegevens geverifieerd zijn. In de zuiverste vorm vraagt het empirisme, dat we vanaf het begin bouwen op bewijsbare, te verifiëren feiten en niet op hypothesen.

Naylor en Finger**) merken op dat een model, waarvan de geldigheid niet door empirische waarneming is vastgelegd geen enkele bijdrage levert tot het begrijpen van het onderzochte systeem.

Een derde stroming zal het geschetste twistpunt hoogst academisch vinden. Voor de aanhangers van deze stroming staat het doel en het nut van het model centraal. Dit zijn de zogenaamde pragmatisten.

Terwijl de rationalisten geïnteresseerd zijn in de interne structuur van het model, gaat het bij de pragmatische vertegenwoordigers van de derde stroming primair om de 'input-output'-relaties. Het simulatiemodel is voor hen een black box; hoe de 'input-output' transformatie in zijn werk gaat, zal hun een zorg zijn. De stelling van de pragmatist luidt: "Zolang het model mijn doelstellingen vervult, is het een valide model". Deze methodologie treffen we wel aan onder de naam 'positive economics'.

Het validatiecriterium van deze richting is de bruikbaarheid, de geschiktheid van het model om het gedrag van de endogene variabelen te voorspellen.

Bekend is het verhaal van het economische model, dat de conjunctuurcycli tracht te verklaren uit het optreden van verhoogde zonne-activiteit (zonnevlekken), die een grote oogstopbrengst ten gevolge zou hebben.

De waarheid ligt gewoonlijk in het midden. Extreme denkbeelden hebben vaak fanatieke volgelingen, die geneigd zijn de problematiek doorgaans eenzijdig te benaderen.

---

*) Zie bijvoorbeeld: J.W. Forrester, Urban dynamics, Massachusets, 1969.

**) T.H. Naylor en J.M. Finger, in: Management Science, "Verification of Computer Simulation Models", Vol. 14, nr. 2, oktober 1967.

Naylor en Finger hebben in het reeds aangehaalde artikel getracht een synthese tot stand te brengen tussen de drie genoemde benaderingen: rationalisme, empirisme en pragmatisme. Elk van de benaderingen geeft een aspect van wetenschappelijk onderzoek. Er is als het ware een integratie van benaderingen noodzakelijk om een onderzoek op verantwoorde wijze uit te voeren.

De bedoelde synthese komt sterk overeen met de aanpak zoals wij die voorstaan en waarvan in de voorgaande hoofdstukken een uiteenzetting is gegeven. Deze aanpak leidt tot wat Naylor en Finger aanduiden als een 'multistage verification process'. In de derde paragraaf van dit hoofdstuk is dit proces behandeld. Als we dit vertalen naar de drie behandelde stromingen in de wetenschapsfilosofie dan leidt dit tot de volgende aanvullingen.

Het eerste stadium van het onderzoek is vooral een meer rationalistische aanpak uitmondend in de verificatie van het model.

Het tweede stadium leidt tot validatie van de interne structuur. Empirische toetsingen vinden plaats op de in de eerste fase geformuleerde hypothesen. Met behulp van diverse statistische technieken worden de verdelingen, de relaties en de parameters aan een onderzoek onderworpen. Of dan de niet bewezen uitspraken uit het model worden weggelaten, dan wel voorlopig(?) toch worden geaccepteerd, is een keuzemogelijkheid voor de onderzoeker.

In de laatste fase wordt het model gevalideerd op zijn vermogen om voorspellingen te doen omtrent het toekomstige gedrag van het reële systeem. Het gebruiksnut van het model staat nu centraal. Nu hebben we te maken met simulatie in de brede zin. We doen uitspraken over de geldigheid van de beeldconstructie. De hoofdvraag is de vraag of het model nuttig is voor de problematiek waarvoor het is opgezet. Het gaat vooral om de acceptatie door de gebruiker en om de implementatie. Praktisch gezien is dan de voorspelkracht erg belangrijk. Het model moet voor de gebruiker nieuwe elementen aanreiken voor de besluitvorming. Het mag niet slechts informatie leveren die de onderzoeker langs een andere, vaak meer eenvoudige weg ook kan verkrijgen.

Voor de analyse van de resultaten staan de onderzoeker diverse technieken ter beschikking. Deze zijn reeds aan de orde geweest. Eén van de middelen om het model goed te evalueren en zodoende het vertrouwen te winnen van de gebruiker is het gebruik van demonstraties. Bij een demonstratie ziet de gebruiker niet alleen het model in 'werking' (in het geval van een interactief model), maar hij kan er ook zelf mee manipuleren onder begeleiding van de onderzoeker.

Concluderend kunnen we stellen dat er voortdurend sprake is van een samenspel tussen de rationalistisch, de empirisch en de pragmatisch georiënteerde visie. Alle drie aspecten komen in het onderzoek aan de orde en worden, als het goed is geïntegreerd.

HOOFDSTUK VII

Decision Support Systems

VII.1 INLEIDING

Dit hoofdstuk gaat over een nieuwe ontwikkeling op het gebied van informatiesystemen: '*Decision Support Systems*'.*) Deze systemen beogen beslissers een ondersteuning te geven bij een bepaald soort beslissingsproblemen, namelijk de zogenaamde zwak gestructureerde problemen. Een toelichting op deze term en op de historische ontwikkeling van informatiesystemen geven we in de volgende paragraaf. Verder zullen we in die paragraaf een aantal van de voornaamste karakteristieken van DSS bespreken.

Met de introductie van DSS is ook een nieuwe aanpak voor de ontwikkeling van informatiesystemen naar voren gekomen die wel wordt aangeduid met de term *prototyping*. Hieronder wordt een ontwikkelingsproces verstaan waarbij het model van het DSS in stappen wordt ontwikkeld. Daar prototyping grote voordelen kan hebben, is het de moeite waard hieraan in dit hoofdstuk (in § 3) enige aandacht te besteden.

Evenals voor simulatiemodellen kan voor DSS de vraag van de toegevoegde waarde worden gesteld. Daar de voordelen vaak alleen kwalitatief zijn, is het moeilijk een kosten/batenanalyse voor deze systemen uit te voeren. In § 4 wordt een *waarde-analyse-methode* behandeld die nuttig lijkt voor de beoordeling van de toegevoegde waarde van een DSS.

In de laatste paragraaf van dit hoofdstuk zullen we ingaan op de samenhang tussen DSS, simulatie en prototyping. De conclusie zal zijn dat prototyping wellicht voordelen biedt voor de ontwikkeling van simulatiemodellen, maar dat met deze methode erg voorzichtig moet worden omgegaan om te voorkomen dat de voordelen teniet worden gedaan door de nadelen.

---

*) Daar een algemeen aanvaarde Nederlandse vertaling nog ontbreekt zullen wij verder de afkorting van de Angelsaksische benaming hanteren, namelijk DSS.

VII.2 ONTWIKKELING EN KENMERKEN VAN DSS

VII.2.1 Drie generaties informatiesystemen

De ideeën rond DSS zijn in de zeventiger jaren met name door Keen en Scott Morton in de publiciteit gebracht. Zij onderscheiden verschillende soorten van geautomatiseerde informatiesystemen, die wij voor het gemak zullen aanduiden als eerste, tweede en derde generatie systemen.

Bij de *eerste generatie informatiesystemen* staat de *automatisering* van *gestructureerde taken* centraal. De ontwikkeling van deze systemen heeft als doel verbetering van de efficiency door kostenverlaging, o.a. door vervanging van administratief personeel. Deze vorm van automatisering heeft betrekking op processen die gekenmerkt worden door standaardprocedures en routinematige beslissingen.

De *informatiesystemen van de tweede generatie* worden gekenmerkt door automatisering van *goed gestructureerde beslissingsproblemen*. Dit zijn beslissingen waarvan de doelstelling éénduidig is en waarvoor stap voor stap kan worden aangegeven wat gedaan moet worden om de doelstelling te realiseren. Dit type van beslissingsproblemen kan in logische stappen worden beschreven en er kunnen zodoende rekenregels (algoritmen) worden gegeven om de doelstelling te realiseren. Een voorbeeld hiervan is een voorraadaanvullingssysteem dat voldoet aan de eisen voor toepassing van de formule van Camp.

Het voordeel van dit soort systemen is dat er betere oplossingen gegeven kunnen worden voor afzonderlijke beslissingsproblemen.

De *informatiesystemen van de derde generatie* omvatten naast automatisering van gestructureerde taken en goed gestructureerde beslissingsproblemen ook modellen voor *zwak gestructureerde beslissingsproblemen*. De computer wordt nu ook gebruikt voor beslissingsprocessen waarbij de beoordeling door het management centraal staat, maar die voldoende structuur bevatten om in de vorm van bijvoorbeeld simulatiemodellen 'op de computer te worden gezet'. Deze ondersteuning wordt aangeduid als beslissingsondersteunend en de systemen die hierop betrekking hebben worden soms wel beslissingsondersteunende systemen genoemd.

We definiëren een DSS als:

> een interactief 'computer-based' systeem dat beslissers met inschakeling van databases en modellen helpt zwak gestructureerde beslissingsproblemen op te lossen.

De nadruk ligt op beslissingsprocessen waarin in ieder geval voldoende structuur aanwezig is om ze in computermodellen af te beelden. De uitkomsten van de processen liggen evenwel niet voor

de hand; de beslisser moet kwalitatieve afwegingen maken en rekening houden met situationele factoren voor de definitief vast te stellen uitkomst. De modellen worden gemaakt om strategieën door te rekenen en zodoende de *effectiviteit* van het beslissingsproces te verbeteren.

Centraal bij de ontwikkeling van een DSS staat ondersteuning van de besluitvorming. De beslisser krijgt met het DSS een instrument in handen dat het beslissingsproces niet automatiseert en dat de beslisser geen oplossingen oplegt. Van essentieel belang in dit verband is, dat het systeem zogenaamde 'wat als'-vragen kan beantwoorden. Hierdoor krijgt de beslisser de mogelijkheid om afhankelijk van de situatie zoals hij die beoordeelt op een bepaald moment een keuze te maken.

VII.2.2 Karakteristieken van DSS

Voor wie en voor welke beslissingen zijn beslissingsondersteunende systemen bedoeld? Aan welke eisen moeten ze voldoen? Op beide vragen zullen we puntsgewijs ingaan. Allereerst noemen we een aantal van de voornaamste karakteristieken van de gebruikers in combinatie met het soort van beslissingen.

a. *Autonome gebruiker*. De gebruiker van een DSS is in hoge mate zelfstandig bij het gebruik van een DSS. Het systeem is vaak sterk georiënteerd op het persoonlijk gebruik. Om deze reden moet de gebruiker op de ontwikkeling van het DSS een belangrijke invloed hebben. De gebruiker weet vaak beter wat hij *niet* wil, dan wat hij *wel* wil.

b. *Zwak gestructureerde problemen*. Met zwak gestructureerde problemen worden problemen bedoeld waarvoor bij voorbaat niet vaststaat welk soort model er moet worden gebouwd. Verschillende oorzaken zijn mogelijk, zoals onduidelijkheid omtrent de nodige informatie, het ontbreken van gegevens of onvoldoende inzicht in de structuur van het probleem. Het te ontwikkelen systeem zal daardoor meestal niet in één keer ontwikkeld kunnen worden, maar in stappen. We noemen dit wel een *incrementeel ontwikkelingsproces*.

c. *Alternatieven*. Een DSS zal antwoord moeten geven op 'wat als'-vragen. Meerdere oplossingen moeten gegenereerd kunnen worden door het systeem op grond van wisselende invoergegevens. Uit deze alternatieven zal de beslisser het voor hem meest bevredigende alternatief als oplossing kiezen. Het begrip *satisfactie* speelt in dit verband een belangrijke rol.

d. *Meerdere doelstellingen*. Door de beslisser wordt vaak een complex van doelstellingen nagestreefd. Het gaat vaak om combinaties: bijvoorbeeld wel maximale winst, maar niet ten koste van arbeidsplaatsen. Verschillende doeleinden moeten door een DSS in hun onderlinge samenhang geanalyseerd worden.

Om deze systemen te realiseren zal aan een flink aantal eisen voldaan moeten worden. *Een deel van deze eisen* heeft betrekking op de technische mogelijkheden van een DSS. We noemen een aantal.

1. *Flexibiliteit.* Beslissingsvraagstukken zijn in de tijd aan verandering onderhevig; de doelstellingen evenals de verklarende variabelen en de relaties kunnen zich wijzigen. Een DSS moet flexibel zijn om gemakkelijk op veranderingen te kunnen inspelen.

2. *Gemakkelijk te gebruiken.* Het systeem moet zonder al te ingewikkelde leerprocessen ingepast kunnen worden in het besluitvormingsproces van de beslisser.

3. *Snel reagerend.* Het systeem mag niet traag zijn. Lange responsetijden beïnvloeden het gebruik in negatieve zin.

4. *Betrouwbaarheid.* Vanzelfsprekend moeten zoveel mogelijk fouten worden voorkomen. De praktijk wijst echter uit dat complexe computerprogramma's vrijwel nooit foutloos zijn. Daarom moeten zowel in de programma's als in de procedures rond de invoer en de beoordeling van de uitvoer waarborgen worden ingebouwd om fouten te voorkomen. We denken hierbij aan *validatieprocedures* met betrekking tot de invoer, de programma's en de uitvoer.

5. *Communicatief.* De kwaliteit van de mens/machine-dialoog en van de uitvoer op het beeldscherm zijn sleutelvariabelen voor een effectief gebruik van DSS. Mensen zullen computersystemen eerder gebruiken als ze passen in hun gebruikelijke manier van werken, als ze geen verstoring maar een ondersteuning blijken te zijn.

VII.3 PROTOTYPING

Met prototyping of het werken met prototypes wordt bedoeld dat bij de ontwikkeling van een DSS wordt gewerkt met leermodellen. Een DSS wordt niet in één keer gebouwd, maar in verschillende versies.

Het woord prototype staat voor 'eerste vorm' of 'eerste model'. Een prototype wordt in de techniek onder meer gebruikt voor testen, het voorbereiden van de produktie of voor demonstratie.

In de toegepaste informatica wordt met prototyping beoogd snel een eerste versie van een DSS te maken. Deze eerste versie kan dan worden gebruikt om de meer volledige specificaties vast te stellen van het volledige DSS. Keen noemt voorbeelden*) van deze twee fasenstrategie. Het ontwikkelen van een bruikbaar prototype blijkt zo'n drie weken tot maximaal enkele maanden te kosten.

Er zijn nog weinig concrete gegevens beschikbaar over de voordelen van prototyping. Men veronderstelt dat prototyping leidt tot:**)

- een hogere betrokkenheid van de gebruikers in een ontwikkelingsproces
- een hogere acceptatie van de systemen
- een kortere ontwikkelingstijd.

Het eerste punt kan worden verklaard uit het feit dat de gebruikers vanaf de aanvang via een incrementeel of evolutionair proces bij de ontwikkeling van het DSS betrokken zijn. Zij worden langs een *'leerpad'* geleid aan de hand van de ontwerper. Langzamerhand zien zij de mogelijkheden van het systeem groeien en raken zo steeds meer betrokken bij het proces.

Met nadruk willen we stellen dat het hier gaat om veel meer dan alleen het leren gebruiken van computers. Van belang is dat de gebruikers steeds meer inzicht krijgen in de beslissingsproblematiek waarmee ze in hun functie worden geconfronteerd. Het model waarmee ze werken kan geavanceerder zijn en het kan daardoor het inzicht in de problematiek vergroten. Met name dit punt past in de door ons aangehangen omschrijving van simulatie als methode van onderzoek. Een simulatiemodel als onderdeel van een DSS kan zo dienen voor *kennisrepresentatie* en *kennisvermeerdering.*

Het tweede genoemde voordeel van prototyping is een logisch gevolg van het eerste punt. Door de hogere betrokkenheid in het ontwikkelingsproces zal, als dit proces goed verloopt ook een hogere acceptatie van het produkt gerealiseerd kunnen worden. De gebrui-

---

*) Zie: Peter G.W. Keen, "Value Analysis: Justifying Decision Support Systems", in: MIS Quarterly, maart 1981.

**) Zie bijvoorbeeld H.G. Sol, "Ervaringen rond prototyping", in: Informatie, jg. 26, nr. 3, maart 1984.

ker zal zich in het produkt herkennen en het zien als een passend onderdeel van het produktieproces.

Het voordeel van de kortere ontwikkelingstijd is betrekkelijk. Inderdaad is het mogelijk, zoals uit de praktijk blijkt, snel een prototype te maken. Zo'n prototype zal daardoor niet duur hoeven te zijn. Het hangt echter helemaal van het vervolg van het project af, of de ontwikkelingstijd korter zal zijn dan bij andersoortige ontwikkelingsprocessen. De duur kan zelfs zeer lang worden als de gebruikers in de beginfase weinig kunnen toevoegen aan het proef-systeem en als de nadere uitwerking slechts in zeer kleine stapjes kan gebeuren. Om te voorkomen dat het geheel te lang gaat duren, kunnen wellicht maatregelen worden genomen. Deze dienen echter met grote zorgvuldigheid te worden toegepast. Het beste is de verantwoordelijkheid in een zo vroeg mogelijk stadium van het ontwikkelingsproject bij de gebruikers te leggen. Door een goede opleiding kunnen ze vaak snel hun eigen produkt onderhouden en verder ontwikkelen.

VII.4 DE TOEGEVOEGDE WAARDE VAN EEN DSS

In § VII.2.1 is opgemerkt dat het er bij DSS vooral om gaat om de effectiviteit van het beslissingsproces te verbeteren. Het gaat vooral om kwalitatieve voordelen, maar rechtvaardigen deze voordelen de kosten?

De basistechnieken, zoals kosten/batenanalyse en toepasbaarheidsstudies zijn niet erg geschikt voor de beoordeling of een DSS verantwoord is in een bepaalde situatie. De belangrijkste voordelen zijn, zoals gezegd, kwalitatief terwijl bovendien de kosten tevoren moeilijk zijn vast te stellen. Voor een mogelijke oplossing van dit vraagstuk heeft Keen de *methode* van de zogenaamde *waarde-analyse* voorgesteld.*) Deze methode volgt de verschillende stappen van de ontwikkeling van een DSS wanneer een incrementeel proces wordt gevolgd. De volgende opzet wordt voorgesteld.

1. Stel de waarde van het te ontwikkelen DSS vast door het opstellen van een lijst van mogelijke voordelen, bijvoorbeeld: het systeem ondersteunt de oplossing van een bedrijfsprobleem, het helpt de planning en de beheersing te verbeteren, het levert tijdsbesparingen op, het geeft snel antwoord op onverwachte situaties, het levert meer alternatieve oplossingen, enzovoorts.

2. Stel een kostendrempel vast aan de hand van schattingen om een eerste prototype te bouwen. Bij de vaststelling van de kosten dient vooral te worden gedacht aan mankracht en apparatuur.

   Probeer de kosten zo laag te houden, dat ze binnen de beslissingsbevoegdheid van de afdeling, waarvoor het DSS is bedoeld, blijven.

3. Bouw de eerste versie van het DSS, ofwel: 'smeed het ijzer als het heet is'. Bij het bouwen van deze eerste versie moeten zoveel mogelijk de technische eisen, die we in § VII.2.2 genoemd hebben, worden aangehouden.

4. Beoordeel het prototype samen met de gebruiker. Ga na of de voordelen in sub 1 omschreven, gerealiseerd zijn en pas zonodig de lijst aan.

Op grond van het gerealiseerde prototype, de voordelen en de kosten hiervan kan worden vastgesteld of het zinvol wordt geacht door te gaan met het project. Als het antwoord negatief is, kan het project worden gestopt. Er zijn dan in ieder geval geen enorme bedragen verloren gegaan. Als het wel zinvol wordt geacht door te gaan, kunnen opnieuw de vier genoemde stappen doorlopen worden.

---

*) Zie: Peter G.W. Keen, ibidem, pag. 11-14.

Het grote voordeel van deze waarde-analysemethode is, dat de te investeren bedragen per fase beperkt kunnen blijven. Terwijl men bij het ontwerp van informatiesystemen volgens het zogenaamde lineaire proces grote risico's loopt, zijn bij het incrementele proces de risico's voor de organisatie beperkt. Bovendien zijn de gebruikers verplicht om de verwachte voordelen tevoren aan te geven. Zij moeten nadenken over de vraag wat ze van het te ontwikkelen systeem verwachten. Hierdoor zullen ze tijdens het ontwerpproces waarschijnlijk een grotere betrokkenheid aan de dag leggen om deze voordelen zoveel mogelijk te realiseren.

VII.5 DSS, SIMULATIE EN PROTOTYPING

In dit boek hebben wij simulatie opgevat als een methode van onderzoek. We bedoelen daarmee een methode waarbij het maken van een model als afbeelding van een reëel systeem centraal staat. Dit model dient om inzicht te krijgen in het gedrag van het systeem, om verschillende strategieën te kunnen evalueren en om het gedrag van het systeem in de toekomst te kunnen voorspellen (zie § II.2). Simulatie is een methode die in belangrijke mate geschikt is voor onderzoek van sociale systemen. Van belang bij dit soort systemen is, dat ze vaak zwak gestructureerd of niet-gestructureerd zijn. Met name voor de zwak gestructureerde systemen kan simulatie een belangrijk hulpmiddel zijn.

Zoals we in de vorige paragraaf hebben betoogd is een DSS een systeem dat met inschakeling van modellen en databases beslissers helpt zwak gestructureerde beslissingsproblemen op te lossen. Een DSS beoogt dus hetzelfde als simulatie. Dit blijkt ook uit de voorbeelden die in de literatuur van DSS worden gegeven: corporate planning, korte-termijn-planning van de produktie en logistiek, vermogensbeheer, capaciteitsplanning, enzovoorts. De conclusie is dat een simulatiemodel een krachtig hulpmiddel kan zijn als onderdeel van een DSS.

In de vorige paragraaf is gewezen op de grote voordelen van een incrementeel ontwikkelingsproces voor een DSS. Deze zelfde voordelen zullen ook voor de ontwikkeling van simulatiemodellen van belang zijn:

- een grote betrokkenheid
- een hoge acceptatiegraad en
- een kortere ontwikkelingstijd.

We kunnen ons nu afvragen of de ontwikkeling van een simulatiemodel zich leent voor prototyping.

Eigenlijk is het antwoord op deze vraag hierboven al gegeven. Simulatiestudies zijn georiënteerd op zwak gestructureerde beslissingsproblemen. Volledig inzicht in dit soort problemen ontbreekt meestal. Dit inzicht kan op verschillende manieren worden vergroot, zoals interview, discussie met experts en literatuurstudie. Naast deze hulpmiddelen is het ook mogelijk eerst een prototype te maken met een beperkt aantal variabelen en relaties. Aan de hand van zo'n prototype, zo'n eerste model kan dan met de gebruiker(s) worden gediscussieerd over de noodzaak van een nadere specificatie. Als zodanig kan prototyping dus een waardevol hulpmiddel zijn om betere simulatiemodellen te ontwerpen.

We willen evenwel naast de mogelijke voordelen ook mogelijke nadelen noemen. Het gevaar van prototyping is ons inziens dat al te snel tot het maken van een model wordt besloten. De redenering kan dan zijn, dat uit het prototype wel zal blijken of het model een redelijke afbeelding is van het reële systeem. Als dan blijkt, dat dit niet het geval is, kunnen in één keer alle voordelen omslaan in nadelen.

Ook bestaat de mogelijkheid, dat een prototype te snel als definitief model wordt geaccepteerd zonder dat voldoende aandacht aan de validatie is besteed. De voordelen van de hoge acceptatiegraad, grote betrokkenheid en korte ontwikkelingstijd gaan dan volledig teniet door het nadeel van de wellicht onjuiste resultaten van het model.

Concluderend willen we stellen, dat prototyping voor simulatiestudies bepaalde mogelijkheden en wellicht voordelen biedt, maar dat met deze methode zeer voorzichtig moet worden omgesprongen. Anders kunnen de voordelen wel eens volledig teniet worden gedaan door de nadelen.

HOOFDSTUK VIII

Besturing van een Simulatieproject

VIII.1 INLEIDING

In dit hoofdstuk wordt o.a. aandacht besteed aan de (na)zorg van een simulatieproject. Het bevat aanwijzingen aan diegenen die zich met modelbouw (gaan) bezighouden. Die aanwijzingen zijn bedoeld om te zeggen: "Let erop, modelbouw is geen louter mathematisch/technische zaak, maar ook een kwestie van succesvolle overdracht aan de gebruikers van het model". Het hoofdstuk kan ook worden gezien als de nazorg door de schrijvers aan de lezers.

We besteden in § 2 aandacht aan het project, aan de voor- en nadelen van deze specifieke organisatievorm in organisaties. § 3 handelt vooral over communicatie en documentatie. Het laatste is vaak een verwaarloosd onderdeel bij projectorganisaties. Als het dan gaat om de overdracht van het produkt, bijvoorbeeld een informatiesysteem of een simulatiemodel, bestaat de beschrijving alleen 'in het hoofd' van een systeemontwerper. Om dit te voorkomen is documentatie noodzakelijk.

In § 4 gaan we in op een aantal punten van misschien wel de moeilijkste fase van het project: de voorbereidingsfase. Dat is de fase waarin de probleemdefinitie en de doelstellingen op tafel moeten komen. Om dit 'rond' te krijgen moet de modelontwerper over geheel andere gaven beschikken dan in latere projectfasen nodig zijn. Een goede voorbereiding is daarom wenselijk. Het is overigens niet uitgesloten om deze fase mee door een ander te laten uitvoeren, bijvoorbeeld een informatie-analist.

Acceptatie en kwaliteit bepalen het succes van een model. Terwijl in de eerste paragrafen op diverse plaatsen aandacht wordt besteed aan de acceptatie, gaat het in § 5 om de kwaliteit van informatie en van het model.

Vanzelfsprekend kan in het kader van dit boek op al deze punten niet uitgebreid worden ingegaan. Daarvoor verwijzen we naar meer algemene literatuur over systeemontwikkeling en projectorganisatie.

Het hoofdstuk wordt besloten met een aantal aanwijzingen en vragen. De aanwijzingen hebben betrekking op de demonstratie van het ontwikkelde model en de vragen op de punten die bij de evaluatie aan de orde moeten komen.

VIII.2 EEN PROJECT: WAT IS DAT EN WAAROM KAN HET FOUT GAAN?

Onder een simulatieproject verstaan we een samenhangend stelsel van activiteiten met als hoofdactiviteiten het ontwerpen, ontwikkelen en implementeren van een simulatiemodel.

Een project heeft een aantal specifieke kenmerken:

- definieerbaar begin- en eindpunt
- duidelijk omschreven doelstellingen
- definitie van de capaciteit die beschikbaar is
- multi-disciplinair van karakter
- eigen organisatie
- eigen budget
- verantwoordelijke leiding in één persoon
- eenmalig van aard.

Behalve simulatieprojecten zijn er op diverse terreinen voorbeelden van projecten aan te wijzen:

- de ontwikkeling van een nieuw produkt
- een bouwproject
- het schrijven van een boek
- de verkiezingscampagne van een politieke partij
- het Apollo-project.

Voor de aanpak van een project is het goed te werken volgens een zogenaamde top-down aanpak. Dit betekent dat vanuit de doeleinden, in globale termen geformuleerd, doeleinden in meer concrete bewoordingen worden afgeleid. Deze concrete doeleinden dienen voor de indeling van verschillende subsystemen waarna deze subsystemen elk afzonderlijk uitgewerkt kunnen worden.

Het succes, of het mislukken van een project hangt o.a. samen met de juiste organisatievorm en de plaats die het project in de organisatie krijgt.

De projectvorm is bedoeld om problemen met een hoge moeilijkheidsgraad buiten de gebruikelijke organisatiestructuur om op te lossen. Dit geldt zeker voor simulatieprojecten maar ook voor decision support systemen, prototypes en informatiesystemen in het algemeen. Hierbij spelen een aantal factoren een rol:

- de onbekendheid van de methode
- de noodzakelijke brede kennis van andere vakgebieden
- de ontwikkelingskosten.

Naarmate de moeilijkheidsgraad toeneemt en het aantal factoren dat hieraan debet is groeit, zal de kans op mislukken groter zijn. Redenen voor het mislukken van projecten staan in het volgende kader bijeen.

Redenen van mislukking van projecten:

1. de basis voor het project is niet gezond
2. onvoldoende steun van directie of bestuur
3. onvoldoende steun van afdelingshoofden
4. verkeerde of geen projectleider
5. onduidelijke taakverdeling
6. geen definitie van begin en einde
7. geen sluitende planning
8. geen bijsturing bij veranderingen
9. veronachtzaming van menselijke aspecten
10. gebrek aan kennis

Het is zinvol deze redenen op een aantal punten nader toe te lichten. We noemen met een korte uiteenzetting de volgende:

- de bemanning: soms zijn ondanks mooie beloften de toegezegde medewerkers niet beschikbaar. Projectparticipatie is dikwijls 'part-time' en heeft niet altijd de hoogste prioriteit bij de deelnemers.
- de opzet: richtlijnen ontbreken, er zijn geen afspraken gemaakt (wat niet is afgesproken, wordt niet gedaan). Er heerst onduidelijkheid t.a.v. de te verrichten taken. Er is geen consistentie door het ontbreken van documentatie.
- het beheer: onvoldoende besturing en begeleiding van het management of een verkeerde projectleider. Een gebrek aan planning.
- de structuur: 'niemand' wil gecontroleerd worden; in het projectteam controleert iedereen elkaar. Leiderschap wordt niet altijd gewaardeerd en niet altijd geaccepteerd. De motivatie is soms afwezig, evenals de noodzakelijke kennis.

Het is in dit verband goed nog eens de aandacht te vestigen op de wetten van Murphy:

- alles wat fout kan gaan, gaat fout;
- alles duurt langer dan men had gedacht;
- alles is duurder dan men had begroot.

Goed projectmanagement betekent het leiden van een project tot het beoogde resultaat met een minimum aan tijd en aan minimum aan kosten.

VIII.3 PROJECTFASERING, COMMUNICATIE EN DOCUMENTATIE

Doel van de projectfasering is het aanbrengen van duidelijke meetpunten waarop de bereikte resultaten worden geëvalueerd en wordt nagegaan in hoeverre de resultaten aan de verwachtingen voldoen. Tevens wordt bekeken in hoeverre de diverse kosten aan de in het budget gestelde normen voldoen.

De fasering van het simulatieproject kan in belangrijke mate overeenstemmen met de fasering zoals die in de voorgaande hoofdstukken is behandeld. Ook in de literatuur, met name in die over de ontwikkeling van informatiesystemen vinden we onder diverse benamingen projectindelingen. Op zich zijn deze benamingen niet belangrijk, het gaat vooral om de taken die in elk van de fasen gedaan moeten worden en om de documentatie die tijdens elk van de fasen moet worden gemaakt.

De taken die in elk van de fasen uitgevoerd moeten worden, hebben we in de hoofdstukken III t/m VI voldoende naar voren gebracht. We willen vooral ingaan op de zin van het maken van documentatie.

Eén van de belangrijkste doelen van documentatie is de *oplossing van het communicatieprobleem*. Dit is vaak het grootste probleem voor mathematisch/technisch geschoolden. In sommige gevallen wordt de communicatiebarrière helaas bewust opgeworpen. Door veelvuldig gebruik van specifiek vakjargon en door te weigeren een documentatiesysteem op te zetten, sluit de specialist zich bewust af van zijn omgeving. Hij realiseert zich niet hierdoor zijn vakgebied in discrediet te brengen.

Soms wordt het jargon als statusverhogend beschouwd: eenvoudige, heldere betogen leggen het qua acceptatie af tegen het veelvuldig gebruik van moeilijke begrippen in combinatie met een academische betoogtrant. Het afschermend effect hiervan is bekend vanuit de medische wereld, maar ook vanuit de informaticawereld. Waar men om een zo groot mogelijke acceptatie te bereiken juist openheid en helderheid zou verwachten, is er sprake van geheimhouding en onduidelijke terminologie.

Om op ons onderwerp terug te komen: als de resultaten van een simulatieproject niet gebruikt worden, is het project, ondanks misschien een hoge kwaliteit van het model mislukt. Kwaliteit èn acceptatie bepalen samen het succes van een simulatieproject.

Behalve om inzicht te geven in het simulatiemodel voor niet-experts is documentatie ook bedoeld als een *middel voor het gebruik en het beheer van het model*. Deze documentatie is meer omvattend dan de eerstgenoemde documentatie. Het gaat hierbij om beschrijvingen t.b.v. systeemontwerpers, programmeurs en operators en om handleidingen voor de eerste lijn-gebruikers van het systeem.

De besproken vormen van documentatie worden wel aangeduid als *systeemdocumentatie*. Een andere vorm van documentatie

is de *projectmanagementdocumentatie*. Deze documentatie heeft ten doel zinvolle informatie te leveren voor de *besturing van een informatieproject*. Hierin komen zaken aan de orde als:

- beschrijving projectorganisatie;
- projectplan en voortgangscontrole;
- tijd- en middelenplan;
- documentatieplan;
- samenvattingen per fase en overzicht van genomen beslissingen.

In geen van beide hierboven genoemde vormen van documentatie wordt gesproken over een beschrijving van het systeem als ontwerp van oplossing. Dit ontwerp maakt deel uit van de zogenaamde *ontwerpdocumentatie*. Voor een simulatieproject kan deze verslaglegging bestaan uit bijvoorbeeld de volgende punten:

- probleemdefinitie en doelstellingenproject;
- modelbeschrijving;
- verantwoording en beschrijving van mogelijke alternatieven;
- experimenteel ontwerp;
- programmabeschrijving;
- validatie;
- resultatenanalyse en voorstellen voor probleemoplossing.

Uit de bovenstaande opsomming van documentatievormen zal duidelijk zijn geworden, dat de uitwerking hiervan veel tijd vraagt. Men leest wel eens (eerder uitzondering dan regel!) dat 30% van de ontwikkelingstijd voor de documentatie gereserveerd wordt. Als deze tijd tevoren, gespreid over het gehele project goed gepland wordt, zal dit niet als storend worden ervaren. Als daarentegen de documentatie achteraf, zoals maar al te vaak helaas het geval is, nog moet worden verzorgd, kan dit soms 'helemaal geen documentatie' tot gevolg hebben.

VIII.4 ENKELE ASPECTEN VAN DE VOORBEREIDINGSFASE

Het allervroegste stadium van de 'geboorte' van een simulatiestudie is vaak moeilijk te achterhalen. Het kan zijn dat de leiding van een organisatie geconfronteerd wordt met een probleem dat zich niet gemakkelijk laat formuleren of waarvan de mogelijke oplossingen zich niet duidelijk aftekenen.

In het algemeen zal gelden dat het probleem qua structuur vrij ingewikkeld in elkaar zit. Goed gestructureerde problemen kunnen vaak door eenvoudige technieken worden opgelost. Simulatie zal echter vooral van toepassing zijn als het gaat om minder goed gestructureerde vraagstukken. Om de kern van zo'n probleem goed in de vingers te krijgen is allereerst overeenstemming over de probleemdefinitie gewenst.

In feite geeft de probleemdefinitie de richting aan waarlangs het modelontwerp wordt opgezet. De aard van het te bouwen model èn de wijze waarop het model kan worden opgelost zijn sterk bepaald door een goede probleemdefinitie. Ook de randvoorwaarden waarmee de grenzen van het probleemgebied worden bepaald, zijn afhankelijk van de probleemdefinitie. Niet voor niets wordt de probleemdefinitie de cruciale fase in het ontwerpproces genoemd.

De opdrachtgever zal in het algemeen in nogal onduidelijke, vage bewoordingen vertellen wat er zijns inziens aan de hand is. Omdat hij het probleem onvoldoende kent, laat staan een oplossing op het oog heeft, vertelt hij welke symptomen hem dwarszitten.

In de eerste verkennende fase waarbij we van de symptoombeschrijving tot de probleemdefinitie willen komen, zal veelvuldig overleg gewenst zijn. Een klassieke fout die wel gemaakt wordt, is om na het eerste gesprek met de opdrachtgever weg te gaan met de boodschap 'Doen we even'. Deze werkwijze zal in een later stadium vrij zeker tot een oplossing van het verkeerde probleem leiden. Niet voor niets circuleren in sommige bedrijven overzichten met de volgende inhoud:

De 6 stadia van een project zijn:

1 enthousiasme

2 ontnuchtering

3 Paniek

4 zoeken naar schuldigen

5 bestraffing van onschuldigen

6 beloning van hen die er niets mee te maken hadden

De moeilijkheden die zich bij de formulering van de probleemdefinitie voordoen hebben nogal eens betrekking op de doelstellingen. Met betrekking tot de doelstellingen geldt, dat deze veelal niet in één richting gaan. Bij de meeste problemen zijn meerdere groepen belanghebbenden aanwezig, elk met eigen doelstellingen. Gedeeltelijk zullen deze doelstellingen elkaar overlappen, maar er zijn ook geschilpunten. Bestudering en bespreking van deze geschilpunten is nodig om tot een algemeen aanvaardbare probleemdefinitie te kunnen komen. Dit is vaak alleen mogelijk als partijen bereid zijn compromissen te sluiten.

Om tot een compromis over de doelstellingen te komen, zijn dikwijls verschillende discussiesessies en interviews nodig. We willen het belang van een goede beheersing van vergader-, discussie- en interviewtechnieken benadrukken. We gaan hier op deze technieken niet nader in.

Voor de doelstellingsanalyse zijn verschillende hulpmiddelen ontwikkeld. Men kan bijvoorbeeld in een bijeenkomst door iedere aanwezige een doelstellingsboom met steeds concreter wordende subdoelstellingen laten opstellen, waarin door elke deelnemer in de vorm van percentages prioriteiten worden aangegeven. Per vergadering probeert men met alle participanten op één lijn te komen.

Men moet niet de vergissing begaan om alle groepen betrokkenen bij elkaar in één bijeenkomst te zetten. Wegens de mogelijk zeer verschillende gezichtspunten is slaande ruzie dan niet uitgesloten.

Een tweede opmerking is, dat men t.a.v. de doelstellingen niet te star moet zijn. Ook in doelstellingen kunnen veranderingen optreden. Men zal met de betrokken groepen moeten overeenkomen dat de doelstellingen herzien moeten kunnen worden (in overleg).

Hiervoor is gesproken over verschillende groepen die bij een (simulatie)project betrokken kunnen zijn. Wie kunnen dit dan zijn, deze groepen?

1. De opdrachtgever

2. De stuurgroep

3. De directe gebruikers van het model

4. Mensen in de organisatie van wie het werk wellicht door implementatie van een nieuw model wordt beïnvloed.

De *opdrachtgever* zal in de organisatie de beslissingsbevoegdheid moeten hebben om het simulatieproject goed te keuren. Zoals reeds eerder gezegd zijn dit soort projecten niet goedkoop. Er zal betrekkelijk veel mankracht, tijd en computergebruik nodig zijn. De voor het project verantwoordelijke leiding zal er achter moeten staan.

Naast de reeds genoemde doelstellingsanalyse is daarom in de eerste fase, ook wel vooronderzoek of toepasbaarheidsonderzoek genoemd, een kosten/batenanalyse (zie ook hoofdstuk VII, § 4) gewenst. De kosten zijn gewoonlijk gebaseerd op de hierboven

reeds genoemde punten als mankracht, d.w.z. verdeling in de tijd van deskundigen belast met de uitvoering en van computertijd.

Vooral de tijd die ontwikkelaars denken nodig te hebben voor de realisatie van een project wordt nogal eens schromelijk onderschat. Ervaren praktijkmensen wijzen erop dat rekening moet worden gehouden met een overschrijding van de tijd met een percentage van 100%.

De baten van een project zijn nog moeilijker te bepalen dan de kosten. Wanneer bij voorbaat aantoonbaar is, dat bij gebruik van een nieuw voorraadbeheersingsmodel het voorraadniveau gemiddeld 10% lager zal zijn bij dezelfde servicegraad is vaststelling van de baten niet zo moeilijk. Doch dit is tevoren niet bekend. Het enige haalbare in dit verband zijn schattingen van ervaren ontwikkelaars op grond van eerder opgedane ervaringen.

De verantwoordelijkheid voor de projectontwikkeling ten aanzien van tijdsduur, kosten en effectiviteit zal door de leiding gedelegeerd kunnen worden aan een *stuurgroep*. In deze stuurgroep zullen zowel experts als belanghebbenden in de organisatie onder leiding van een voor elke groep aanvaardbare voorzitter zitting hebben. De keuze van de voorzitter is van essentieel belang voor een goede uitwerking van het project en voor een succesvolle implementatie van een (eventueel) nieuw systeem. De voorzitter moet iemand zijn met zowel materiedeskundigheid als met kennis van de toegepaste methode. Hij moet het vertrouwen hebben van de deelnemers en vanzelfsprekend van de opdrachtgever.

Zoals in het vorige hoofdstuk al behandeld is, zullen simulatiemodellen een onderdeel vormen van een DSS of een management informatiesysteem (MIS). Vooral ten behoeve van planningsactiviteiten zullen simulatiemodellen hun diensten kunnen bewijzen. Ze bieden, als het goede modellen zijn, de mogelijkheid alternatieve oplossingen door te rekenen en zogenaamde 'wat-als'-vragen interactief te beantwoorden. Om van deze mogelijkheden een effectief gebruik te kunnen maken, zullen de modellen door de *directe gebruikers* geaccepteerd moeten worden. Dit zijn dus de gebruikers die geacht worden de informatie van het model te betrekken bij hun beslissingen. Afhankelijk van het toepassingsgebied kunnen dit financiële, personele, commerciële, logistieke of produktiemanagers zijn.

De laatste categorie *betrokkenen* is de groep *wier werk* door de implementatie van het nieuwe model *wordt beinvloed*. Het kunnen bijvoorbeeld mensen zijn die de gegevensvoorbereiding moeten verzorgen of die voor een foutloze technische uitvoering van de programma's zorg moeten dragen. De betrokkenheid van deze groepen is van even groot belang als van andere, op het eerste gezicht misschien belangrijker mensen. Veelal zullen deze laatsten geen tijd kunnen besteden aan bijvoorbeeld het onderhoud van het model. Ze zijn daarvoor afhankelijk van meer technisch geöriënteerd personeel. Als deze groep zijn taak niet goed uitvoert, zou dit wel eens tot een beëindiging van het modelgebruik kunnen leiden.

We laten tenslotte, zonder verder commentaar, een aantal punten volgen waaraan in deze fase van het onderzoek aandacht moet worden besteed:

1. het probleem en de doelstellingen moeten in duidelijke termen beschreven worden. Hierover mag geen verschil van opvatting bestaan;

2. er moet een kostenbudget worden gemaakt alsmede een tijdplanning, een personele planning en hulpmiddelenplanning;

3. vaststelling van het projectmanagement en van een stuurgroep;

4. het ontwikkelen van criteria aan de hand waarvan de resultaten getoetst worden;

5. vaststelling van de categorieën medewerkers die bij het project betrokken zijn en van de wijze waarop ze meedraaien in het project.

VIII.5 KWALITEITSASPECTEN

In de vorige paragrafen is o.a. gesproken over zaken die de acceptatie van het model ondersteunen. Vanzelfsprekend zal de acceptatie in de eerste plaats beïnvloed worden door de kwaliteit van het model. Het oordeel van de modelbouwers is hierbij van minder belang dan dat van de uiteindelijke gebruikers.

De kroon op het werk is een succesvolle modelimplementatie. Zolang het model niet daadwerkelijk door de belanghebbenden wordt gebruikt, kan niet van een geslaagd project worden gesproken. De gebruikers zijn niet geïnteresseerd in de elegantie van het model of in de gebruikte technieken, hoe geavanceerd deze wellicht ook zijn. Het gaat hun om de kwaliteit van de informatie die als ondersteuning bij de besluitvorming kan worden gebruikt. Niet het model zelf is belangrijk, maar de uitvoer. Deze zal moeten voldoen aan de criteria van de gebruiker.

We noemen enkele kwaliteitsaspecten van informatie die tijdens het project besproken moeten worden en die voor de gebruiker als toetsingscriteria beschouwd kunnen worden.

- Relevantie: d.w.z. dat de informatie zo goed mogelijk moet zijn afgestemd op het doel en het subject voor wie ze bestemd is.

- Betrouwbaarheid: de verstrekte informatie mag geen fouten, bijvoorbeeld als gevolg van fouten in de invoergegevens of in de verwerking bevatten. Daarom is controle noodzakelijk.

- Nauwkeurigheid: de informatie dient zo nauwkeurig te zijn als voor het gebruiksdoel noodzakelijk is. Dit punt hangt samen met het aspect van de relevantie.

- Bruikbaarheid: dit punt heeft betrekking op de presentatievorm. De informatie moet zo gepresenteerd worden, dat bewerkingen door de ontvanger niet meer nodig zijn. De informatie moet overzichtelijk zijn; hiervoor kan gebruik worden gemaakt van bijvoorbeeld tabellen en grafieken.

De kwaliteit van de informatie hangt samen met de kwaliteit van het model. We laten enkele eisen, die in dit verband aan een model gesteld kunnen worden, volgen. Omdat een zekere strijdigheid tussen deze eisen niet is uitgesloten, is een afweging in een vroeg stadium noodzakelijk.

- Het model moet in het gebruik zijn waarde bewijzen om zodoende de kosten van ontwikkeling en onderhoud te kunnen motiveren. Hoewel een schatting van de diverse componenten niet gemakkelijk is, zal het model getoetst moeten worden op basis van kosten en opbrengsten.

- Het model moet begrijpelijk en gemakkelijk hanteerbaar zijn. Moeilijke leerprocessen zullen de gebruiker afschrikken. Gebruikersvriendelijke, bij voorkeur interactieve modellen met 'hulpfuncties', een plezierige dialoog en zelfbeslissing m.b.t. vormen van uitvoer zijn belangrijke zaken voor acceptatie van het model.

- Het model moet robuust zijn, d.w.z. dat zelfs bij invoering van niet gebruikelijke, afwijkende invoergegevens het model moet werken. De programmatuur moet 'bestand' zijn tegen abnormale invoer.

- Het model moet zoveel mogelijk aansluiten bij de alledaagse gang van zaken. Dit betekent bijvoorbeeld t.a.v. de invoergegevens dat deze redelijk gemakkelijk beschikbaar moeten zijn in de organisatie. Shannon noemt een voorbeeld van een ziekenhuismodel dat nooit gebruikt werd vanwege de vele administratieve procedures die nodig waren om de noodzakelijke patiëntengegevens te kunnen leveren. In het algemeen moet het gebruik van het model aansluiten bij de realiteit, geen confectie maar maatwerk.

- Het model moet antwoord kunnen geven op 'wat-als'-vragen. Dat hangt samen met het reeds genoemde feit, dat de gebruiker met het model zal willen 'spelen'. Hij zal aan het model in een interactieve sessie vragen willen stellen, zoals bijvoorbeeld:
  "Welke situatie ontstaat er, als de verkoop daalt met x% en de produktie doorgaat met een bezettingsgraad van y%?"
  Met dergelijke sessies past de gebruiker in feite een gevoeligheidsanalyse toe ten behoeve van zijn planningstaak. Hij probeert de onzekerheid te schatten en ten aanzien van de gewenste toekomstige toestand de relevante factoren te vinden.

- Het model dient flexibel te zijn, dat wil zeggen gemakkelijk aan te passen aan gewijzigde omstandigheden of uit te breiden. Daartoe is in ieder geval een goede documentatie nodig, maar ook een flexibele programmastructuur.

Het is duidelijk dat deze eisen een sterke overeenkomst (moeten) vertonen met de eisen die aan DSS zijn gesteld. Het is van belang dat de in meer algemene termen geformuleerde eisen aan de specifieke toepassingen van simulatie, DSS of informatiesystemen moeten worden aangepast.

VIII.6 SLOT

Simulatiemodellen worden (gelukkig) in steeds bredere kring gebruikt. In het algemeen zullen de belanghebbenden tijdens het project, bijvoorbeeld als lid van de stuurgroep voldoende betrokken zijn geweest en met het resultaat niet onbekend zijn. Toch zal met het oog op de acceptatie nog veel afhangen van de definitieve verschijningsvorm van het model. Door middel van een presentatie in de vorm van een demonstratie zal de gebruiker definitief 'gewonnen' moeten worden. Aan dit onderdeel zal daarom aandacht besteed moeten worden.

Hieronder volgen een aantal aanwijzingen:

1. verzorg de presentatie met behulp van aantrekkelijke hulpmiddelen, zoals dia's en terminals. Verstrek geen omvangrijke rapporten, maar goed leesbare en overzichtelijke uittreksels;

2. laat de mondelinge presentatie van het model niet te lang duren. Laat de gebruikers zo snel mogelijk zelf oefenen op beschikbare terminals;

3. gebruik geen vakjargon. Dit vraagt om een goede voorbereiding want experts passen hun taalgebruik niet zomaar aan;

4. wees eerlijk t.a.v. de mogelijkheden van het model. Vertel de doelstellingen en laat zien in hoeverre deze gehaald zijn.

Als dan tenslotte het model geïmplementeerd is en in gebruik is genomen, volgt nog de evaluatie en de nazorg. Het doel van de evaluatie is om die ervaringen uit het simulatieproject te halen, die ons in staat stellen in het vervolg betere systemen te ontwikkelen.

Bij de evaluatie moeten de volgende voor de hand liggende vragen worden gesteld:

- hoe worden de projectdoelstellingen achteraf ervaren?

- zijn de doelstellingen gehaald?

- in welke mate is het model verankerd in de organisatie en geaccepteerd?

- hoe zijn de omstandigheden van de gebruikers beïnvloed?

- hoe is het project verlopen; zijn de kostenschattingen en de planningen juist gebleken?

- hoe is de samenwerking verlopen?

- welke problemen zijn nog niet opgelost?

Deze laatste vraag hangt samen met de nazorg. Tekortkomingen zullen vaak pas na verloop van tijd aan de dag komen. Deze komt men sneller op het spoor als het contact met de gebruikers wordt onderhouden. Als men in staat is hierop adequaat in te spelen, zal dit het vertrouwen in modelgebruik doen toenemen. En daar gaat het uiteindelijk om, want:

> "De essentie van de informatiemaatschappij is niet dat er steeds meer computers komen, maar dat steeds meer delen van de werkelijkheid worden nagebouwd in wiskundige modellen waarmee men op computers dingen kan uitrekenen".*)

---

*) Fred Kappetijn, in: Intermediair, 'De ontvanger bepaalt zelf wel wat informatie is', jg. 17, nr. 25, juni 1981.

HOOFDSTUK IX

Cases

A. Ontladen
B. Ziekenhuis

INLEIDING

We hebben reeds een aantal malen naar voren laten komen dat simulatie-onderwijs zonder praktische oefening niet zinvol is. Simulatie is alleen te leren als studenten hun tanden eens kunnen zetten in een opgave waarbij alle facetten van het onderzoek aan de orde komen. Afhankelijk van de beschikbare tijd kan de praktische opgave variëren van ingewikkeld, bijvoorbeeld een opdracht waarbij alleen wordt gewezen op een probleem in een bepaalde concrete situatie en aan de studenten wordt gevraagd het gehele onderzoek uit te voeren, tot eenvoudig in die gevallen dat de probleembeschrijving met alle data gegeven wordt en de modelbouwfase al min of meer 'rond' is. We wijzen er overigens op, dat ook voor eenvoudige opgaven vrij veel tijd nodig blijkt te zijn. We gaan er dan wel vanuit dat de studenten zelf een computerprogramma moeten schrijven en een verslag van het onderzoek aanbieden. Voor meer ingewikkelde opgaven, zoals het in dit hoofdstuk beschreven probleem van de ziekenhuissimulatie moet aan een werktijd van 2 à 3 maanden worden gedacht. Daarbij is het bovendien nog zo dat de opdrachten in groepjes van minimaal twee personen worden uitgevoerd.

Het werken in kleine groepjes lijkt ons om verschillende redenen zinvol:

- tijdsoverweging (zie hierboven)
- samenwerkingsoverweging:
  de studenten leren zoals ook in de praktijk gebruikelijk is, samen een project op te zetten, uit te werken, taken te verdelen en verantwoor verantwoordelijkheid te dragen;
- de veelheid van de aspecten:
  zoals uit het boek blijkt moeten studenten veel onderwerpen beheersen om een simulatie-onderzoek in al zijn facetten te kunnen uitvoeren; de groepssamenstelling moet daarom zodanig zijn dat de diverse aspecten van het onderzoek over de groepsleden gespreid zijn.

Er zijn in dit boek twee case-beschrijvingen opgenomen, die gebaseerd zijn op de onderzoekverslagen van studenten.

De eerste case, 'Ontladen', is vervaardigd door een drietal studenten van de Interfaculteit Bedrijfskunde te Groningen in het kader van het tweede-jaars vak 'Kwantitatieve besluitvormingstechnieken'.De tweede case betreft een eindexamenverslag van twee studentes aan de HEAO-BI (Bedrijfskundige Informatica) te Den Haag voor het derde-jaars vak 'Simulatie'. De verslagen zijn hier en daar ingekort. Dit geldt met name voor de programmadocumentatie. De computerprogramma's hebben wij niet opgenomen.

Onze keus is op deze twee verslagen gevallen, omdat ze beide vrij volledig zijn in de onderwerpen die aan bod dienen te komen: modelbeschrijving, keuze van de verschillende soorten variabelen, experimenteel ontwerp, programmabeschrijving, validatie en resultatenanalyse.

De beide verslagen zijn *niet* gekozen, omdat ze foutloos zouden zijn. Maar er staan niet zodanige fouten in dat aan de conclusies van het onderzoek behoeft te worden getwijfeld.

BESCHRIJVING VAN DE CASE 'ONTLADEN'

Bij het centraal magazijn van een groot warenhuis komen regelmatig vrachtauto's aan. Deze worden ontladen door heftrucks met chauffeurs. Het bedrijf wil onderzocht hebben met welk aantal heftrucks de kosten, verbonden aan het ontladen, geminimaliseerd worden. Deze kosten bestaan uit gewoon loon, eventueel overwerkloon (beide t.a.v. de heftruckchauffeurs) en de wachtkosten voor de vrachtwagens.

Uit eerder onderzoek zijn de volgende gegevens bekend:

- de aankomst van de vrachtauto's kan beschreven worden d.m.v. een negatief exponentiële verdeling met een gemiddelde tussenaankomsttijd (g.t.t.) van 2 uur. De auto's komen alleen tussen 8 en 17 uur aan;
- de tijd nodig voor het ontladen van een vrachtauto is afhankelijk van de soort lading en het gewicht van de lading;
- de kansverdeling van de soort lading ziet er als volgt uit:

soort lading	kans op deze lading	ontladingssnelheid (kg/uur)
A	0.40	8000
B	0.35	7000
C	0.25	5000

- het gewicht van de ladingen is normaal verdeeld met een gemiddelde van 30.000 kg en een standaarddeviatie van 4000 kg;
- de wachtkosten voor een vrachtwagen zijn $ 30,- per uur;
- de gewone loonkosten van een heftruckchauffeur zijn $ 20,- per uur;
- de overwerkloonkosten van een heftruckchauffeur zijn $ 40,- per uur.



UITWERKING VAN DE CASE ONTLADEN*)

Hoofdstuk 1 DE PROBLEEMDEFINITIE

1.1 DOELSTELLING

Het geschetste probleem zal door middel van simulatie onderzocht en opgelost worden. We kunnen de doelstelling van dit simulatie-onderzoek als volgt formuleren:

> Het onderzoek van het gedrag van het systeem om te kunnen bepalen wat het meest geschikte aantal in te zetten heftrucks is bij een aanvaardbaar niveau van wachttijden tegen zo laag mogelijke totale kosten.

Het gaat hierbij dus om een afstemmingsprobleem tussen verschillende beslissingsfactoren.

De gegevens die uit het simulatiemodel naar voren moeten komen zijn de gemiddelde totale bedrijfskosten per dag bij het gebruik van x heftrucks (x = 1,2,...,n) en de daarbij behorende overwerktijd.

1.2 HET SYSTEEM

ELEMENTEN:

Tot de belangrijkste elementen die van belang zijn voor het systeem behoren:

- de binnenkomende vrachtwagens,
- de lading van deze auto's (soort en gewicht),
- de heftrucks,
- de werktijden van de heftruckchauffeurs,
- het ontladingsbeleid en
- de wachttijd van de vrachtauto.

SYSTEEMGRENZEN:

Het systeem dat onderzocht wordt is in eerste instantie beperkt tot het warenhuis zonder omgevingsinvloeden. Daarvan wordt alleen het subsysteem m.b.t. het ontladen van vrachtwagens bekeken en dan alleen nog maar de aspecten gemiddelde totale kosten en gemiddelde overwerktijd.

UITGANGSPUNTEN:

Voordat we dit systeem in een model omzetten hebben we eerst een aantal uitgangspunten geformuleerd:

---

*) De uitwerking van de case 'Ontladen' is gebaseerd op het werkstuk van H. Matthijsen, A. Nysingh en C. de Vet, studenten aan de Interfaculteit Bedrijfskunde te Groningen, september 1983.

- de normale werktijd van 8 tot 17 uur is 9 uur, er wordt continu doorgewerkt;
- elke vrachtwagen wordt door 1 heftruck gelost;
- een vertrekkende vrachtwagen die aan ziet komen dat hij niet voor 17 uur zal aankomen vertrekt niet meer die dag, maar pas de volgende dag (de betekenis voor ons model is dat alle gegenereerde tussenaankomsttijden gebruikt worden);
- de vrachtwagen die het eerst binnenkomt wordt het eerst geholpen;
- zodra de heftruck klaar is met het ontladen van een auto verdwijnt deze auto van het terrein (i.v.m. de wachtkosten);
- heftruckchauffeurs zijn altijd beschikbaar, ook voor overwerk;
- de heftruck die het eerst beschikbaar is, neemt de vrachtauto die het eerst is aangekomen;
- tijdens het ontladen wordt er niet van heftruck gewisseld;
- de kosten van heftrucks worden niet meegenomen, alleen de loonkosten van de chauffeurs;
- onderhoud en storingen, evenals andere afwijkende situaties worden buiten beschouwing gelaten;
- verschillende ontladingstijden beïnvloeden elkaar niet;
- aankomsttijden zijn onafhankelijk van elkaar.

Hoofdstuk 2 MODELFORMULERING

Het model is bestemd voor het verkrijgen van gegevens die de directie van het warenhuis in staat stellen het optimale aantal heftrucks voor de gegeven situatie te bepalen.

2.1 PARAMETERS

De belangrijkste parameters die we in het model kunnen onderscheiden zijn:

- de gemiddelde tussenaankomsttijd (van de vrachtwagens),
- gemiddelde en standaarddeviatie van het gewicht van de lading,
- de kansverdeling van de soort lading plus de bijbehorende ontladingssnelheden.

2.2 VARIABELEN

2.2.1 Exogene variabelen

De exogene variabelen zijn te splitsen in omgevings- en instrumentele variabelen.

Omgevingsvariabelen zijn:
1. aankomsttijd vrachtwagens
2. gewicht lading
3. soort lading (en ontladingssnelheid)
4. wachtkosten
5. loonkosten

De instrumentele- of beslissingsvariabelen zijn:
6. aantal heftrucks
7. ontladingspolitiek
8. de maximale toegestane overwerktijd van de heftruckchauffeurs.

2.2.2 Endogene variabelen

De endogene variabelen zijn te splitsen in toestands- en uitgangsvariabelen.
We hebben als toestandsvariabelen:
9. tijdstip op de dag
10. lengte van de wachtrij op elk moment

De uitgangsvariabele is:
11. de totale kosten per dag.

2.3 RELATIES IN HET MODEL

In onderstaand schema is de invloed van de belangrijkste variabelen op elkaar weergegeven.

	overwerktijd	totale kosten
aantal heftrucks	neg.	pos./neg.
ontladingstijd	pos.	pos.
tussenaankomsttijd	neg.	neg.

2.4 DE RELATIES IN EEN MODELSCHEMA

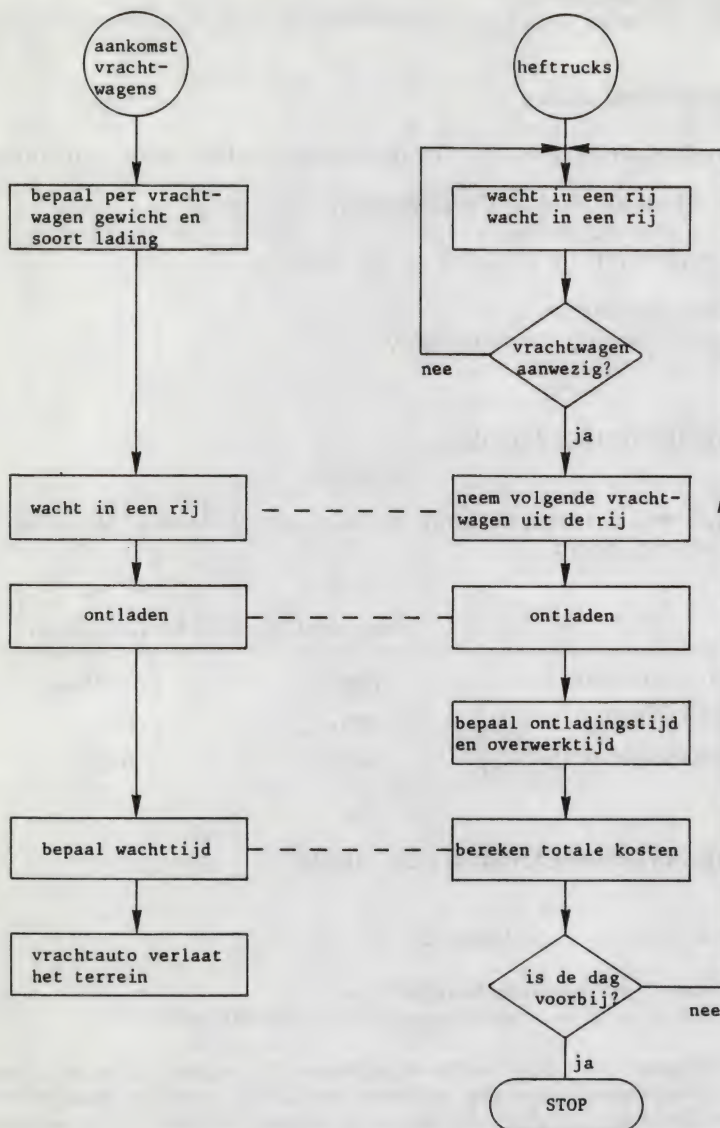
Het model is op te splitsen in:

a. de aankomst van de vrachtwagens en
b. de ontlading van de vrachtwagens door de heftrucks.

De aankomst van de vrachtwagens wordt bepaald door de tussenaankomsttijden die gegenereerd worden met een gemiddelde tussenaankomsttijd van 2 uur. De ontladingen worden in sterke mate bepaald door het aantal heftrucks en de ontladingssnelheid. Deze laatste is op zijn beurt weer afhankelijk van het gewicht en de soort lading welke de betreffende vrachtwagen transpor-

teert. Verder beïnvloedt het eerste deel het tweede deel, doordat als de tussenaankomsttijden klein zijn de heftrucks vaker bezet zijn en de wachttijden voor de vrachtauto's toenemen.

Het model kan men globaal weergeven met het model- of processchema, afgebeeld op de volgende bladzijde. De in de gestippelde lijnen weergegeven relaties zullen in het programma-ontwerp expliciet gemaakt worden.



Proces-schema: Globale weergave van het te bouwen model

2.5 VALIDATIE VAN HET MODEL

De validatie van een model wordt bepaald door de mate waarin het gedrag van het model overeenkomt met het reële systeem, ofwel:

- zijn alle relevante exogene variabelen opgenomen?
- zijn alle relevante relaties gelegd?
- zijn de relaties juist geformuleerd?
- zijn de schattingen van de parameters reëel?

De grenzen waarbinnen een model valide genoemd wordt, zijn afhankelijk van de doelstelling van het onderzoek. Voor dit onderzoek zijn alleen de directe kosten, zoals loon-, overwerk- en wachtkosten van belang.

De in hoofdstuk 1 gestelde hypothesen t.b.v. de bouw van het model dienen getoetst te worden aan de werkelijkheid om een uitspraak te kunnen doen over de validiteit van het model. Vanwege het ontbreken van gegevens omtrent deze werkelijkheid beschouwen we de gestelde hypothesen als zijnde in overeenstemming met de werkelijkheid en laten we een verdere toetsing hiervan in dit rapport achterwege.

Ook de in het model gebruikte verdelingen en parameters dienen als valide aangenomen te worden vanwege het ontbreken van toetsingsmogelijkheden.

De consistentie van het model is het enige dat we kunnen controleren, namelijk door het modelschema en het stroomdiagram van het computerprogramma een aantal malen door te lopen.

Hoofdstuk 3 EXPERIMENTEEL ONTWERP

Het doel van het experimenteel ontwerp is om de gewenste experimentele data op de meest efficiënte manier ter beschikking te krijgen. Experimenteel ontwerp bestaat uit twee fasen:

1. een strategische planning die de opzet van het systeem bepaalt en
2. een tactische planning die zich bezighoudt met de vraag: hoe voer ik het uit?

3.1 STRATEGISCHE PLANNING

Onder de strategische planning vallen het structureel en het functioneel model.

In het structurele model bepalen we welke factoren in het simulatiemodel gevarieerd moeten worden. De belangrijkste factor in ons model is het aantal heftrucks, welke in principe oneindig groot kan zijn. Een andere factor die van belang kan zijn is de ontladingspolitiek. De werktijden worden als vast gegeven beschouwd, maar zouden ook als factor meegenomen kunnen worden.

Het functionele model is een uitwerking van het structurele model. Het functionele model geeft aan welke factoren en welke waarden uiteindelijk bij het simuleren gebruikt worden. De ontladingspolitiek is reeds vastgelegd in onze uitgangspunten en zal verder niet gewijzigd worden. De enige niet te variëren factor die dan overblijft, is het aantal heftrucks. In eerste instantie hebben we het maximale aantal heftrucks op 6 gesteld, een willekeurige keuze om het model te bestuderen. Later hebben wij dit aantal uitgebreid tot 8, omdat daardoor het optimale aantal heftrucks duidelijker naar voren kwam. Het aantal meetpunten komt hierdoor op 8.

3.2 TACTISCHE PLANNING

Voor de tactische planning zijn de volgende zaken van belang:

- WAARNEMINGSPERIODE:
  een werkdag van 9 uur zonder pauzes.
- RUNLENGTE:
  wordt bepaald door de steekproefomvang. We maken gebruik van de voortgezette runmethode.
- STEEKPROEFOMVANG:
  een nauwkeurige bepaling hiervan volgt hierna.
- STARTWAARDE:
  elke dag wordt met een leeg systeem begonnen. De laatste waarde van de vorige dag die niet gebruikt is, wordt bewaard en is de eerste waarde van de volgende dag.
- UITVOERANALYSE:
  over de gesimuleerde dagen krijgen we de volgende gegevens:
  - gemiddelde totale kosten per dag voor respectievelijk 1,2,3,4,5,6,7 of 8 heftrucks,
  - de hierbij behorende standaarddeviatie,
  - gemiddelde overwerkminuten per dag voor respectievelijk 1,2,3,4, 5,6,7 of 8 heftrucks met de daarbij behorende standaarddeviatie,
  - het gemiddelde van de tussenaankomsttijden en de standaarddeviatie,
  - het gemiddeld gewicht van de ladingen met de standaarddeviatie
  - de gemiddelde ontladingssnelheid met de standaarddeviatie.

De hierboven vermelde gegevens zijn die gegevens die we uiteindelijk nodig hebben. In de testfase van ons computerprogramma is ook gewerkt met een tussenuitvoer, waarin per dag de aankomsttijden, het gewicht, de ontladingstijd, de diverse kosten enzovoort afgedrukt werden. Voor een voorbeeld hiervan verwijzen we naar bijlage 2.

DE BEPALING VAN DE STEEKPROEFOMVANG

Voor de bepaling van de steekproefomvang maken we gebruik van de volgende formule voor normale verdelingen:

$$n = \left(\frac{k \cdot s \cdot 100}{b \cdot \bar{x}}\right)^2$$, waarin:

k = een waarde gerelateerd aan de mate van betrouwbaarheid. Bij een 95% betrouwbaarheidsinterval is $k = 1,96$,
s = de standaarddeviatie,
$\bar{x}$ = het gemiddelde,
b = de procentuele afwijking van het gemiddelde van de steekproef t.o.v. het gemiddelde.

In ons onderzoek hebben we te maken met drie verdelingen, namelijk:

A. een negatief exponentiële verdeling met een g.t.t. = 120 minuten, zodat $\lambda = 1/120$ en $E(X) = 1/\lambda$ en $V(X) = 1/\lambda^2$

B. een normale verdeling met $\mu = 30.000$ kg en $\sigma = 4000$ kg

C. een histogram als volgt:

kans	waarde (ontladingssnelheid in kg/h)
0,40	8000
0,35	7000
0,25	5000

Volgens de centrale limietstelling mag je aannemen dat bij voldoende grootte van de steekproef het gemiddelde van de steekproef bij benadering normaal verdeeld is. Dus ook voor de kansverdelingen A en C kan bij voldoende grootte gebruik worden gemaakt van bovenstaande formule.

Als we b op 10% stellen zou voor de negatief-exponentiële verdeling een steekproef met een omvang van $n = 385$ voldoende zijn. Voor een grotere nauwkeurigheid zou de steekproefomvang nogal wat groter worden, hetgeen ons niet nodig lijkt.

Ten aanzien van de kansverdelingen B en C blijkt de b bij een n van 385 erg klein te zijn:

B: b = 1,3%
C: b = 1,9%

De afwijkingen t.o.v. het gemiddelde zien er bij een steekproef met $n = 385$ als volgt uit:

A: $108 \leq \underline{\bar{x}}_A \leq 132$ $\quad \mu_A = 120$

B: $29.610 \leq \underline{\bar{x}}_B \leq 30.390$ $\quad \mu_B = 30.000$

C: $6.769 \leq \underline{\bar{x}}_C \leq 7.031$ $\quad \mu_C = 6.900$

Met het aantal trekkingen van 385 was nog niet het aantal te simuleren dagen bekend. Een schatting van 90 dagen bleek te laag. Een tweede schatting van 110 dagen bleek iets te hoog (400 trekkingen), maar om het probleem van halve dagen te voorkomen, alsmede vanwege het mooie ronde getal, hebben we het aantal te simuleren dagen op 110 laten staan.

Hoofdstuk 4 PROGRAMMASTRUCTUUR EN VALIDATIE

4.1 PROGRAMMASTRUCTUUR

De taak van het programma is, om uitgaande van de aankomsttijd van de vrachtwagens en het aantal heftrucks, het ontladingssysteem een aantal dagen te simuleren en gegevens m.b.t. de gemiddelde kosten en de gemiddelde overwerktijd per dag te berekenen voor een steeds verschillend aantal heftrucks.

Het programma is te verdelen in vier blokken die voor iedere te simuleren dag doorlopen moeten worden. Deze vier blokken zijn:

1. bepaling van de aankomsttijden van de vrachtwagens;
2. bepaling van de bij de vrachtwagens behorende ontladingstijden;
3. ontlading van de vrachtauto's door een verschillend aantal heftrucks;
4. statistische verwerkingen.

In blok 3 zijn nog twee onderdelen te onderscheiden, namelijk:

3a. bepaling van de heftruck die het eerst beschikbaar is voor de ontlading van de volgende vrachtauto en

3b. berekening van de totale kosten en de tijd dat overgewerkt wordt per dag.

Ten behoeve van de statistische bewerkingen in blok 4 worden reeds in de blokken 2 en 3 enkele berekeningen uitgevoerd.

De structuur van het programma is gegeven in het programmastructuurdiagram (bladzijde 202). In het programma is nog de mogelijkheid aanwezig om de eerder genoemde tussenuitvoer, die bestaat uit relevante gegevens per dag, uit te laten schrijven.

Het programma kan veel gebruikersvriendelijker gemaakt worden dan het nu is. Dit vonden wij echter niet nodig, alleen al vanwege het beperkte gebruik.

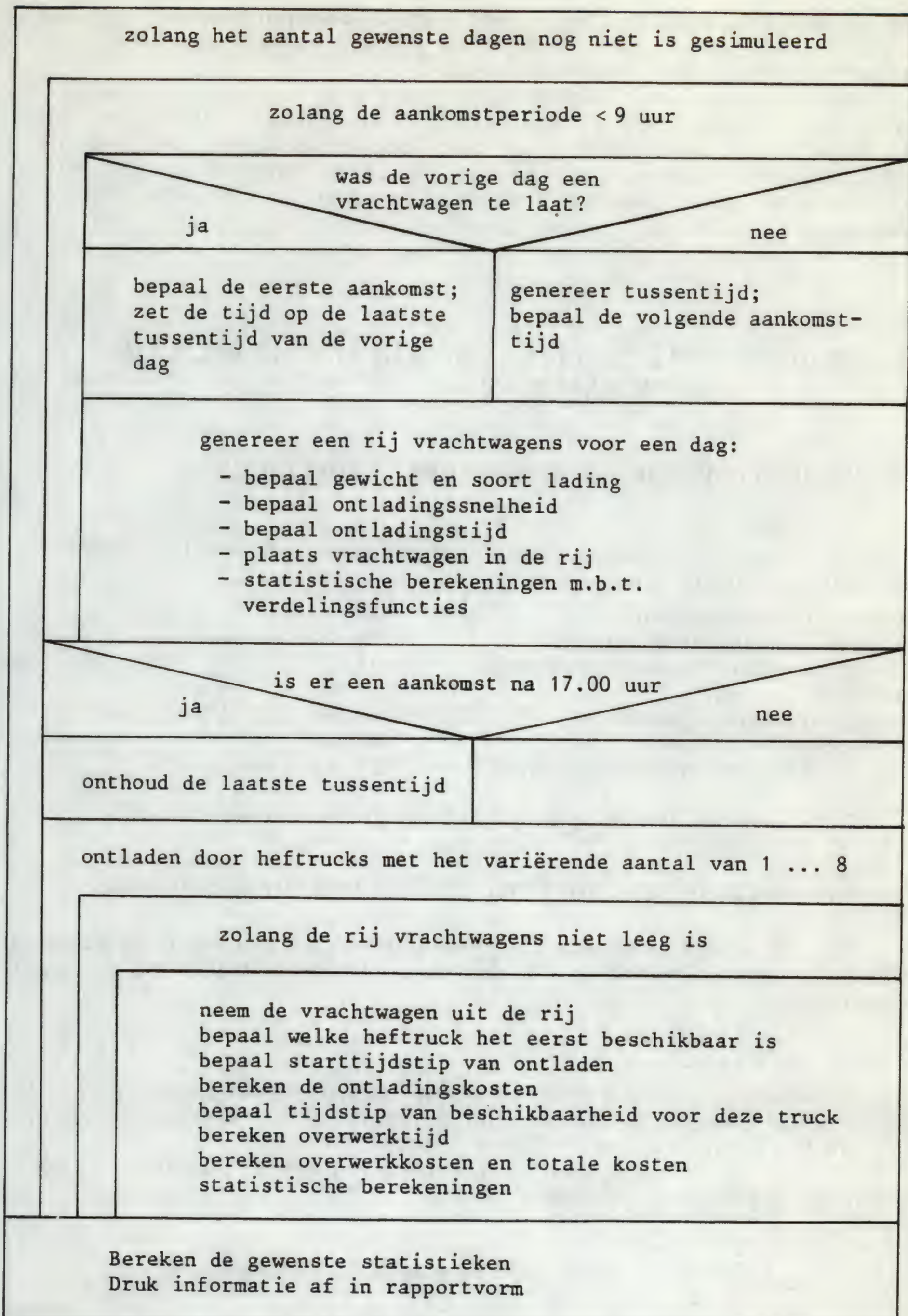
4.2 VALIDITEIT VAN HET COMPUTERMODEL

Het toetsen van het computermodel bestaat uit de controle of het computerprogramma wel een 1:1-afbeelding is van het eerder ontworpen model. De toets is te verdelen in 3 soorten analyses, te weten:

- een syntactische analyse,
- een semantische analyse en
- een pragmatische analyse.

Syntactische fouten worden door de computer 'automatisch' opgespoord en gemeld. Vanzelfsprekend zijn dergelijke fouten meermalen voorgekomen. Denk alleen maar aan de vele typefouten en slordigheden.

Semantische fouten kunnen voor een deel door de computer worden opgespoord. Dat is eenmaal voorgekomen en betrof een onvoorziene deling door nul. Andere semantische fouten die werden ontdekt bij het testen van



Programmastructuurdiagram 'Ontladen'

het programma waren een niet correct ingevoerde formule en het niet meenemen van een dermate grote tussentijd, dat een dag overgeslagen wordt, bij het berekenen van het gemiddelde.

Pragmatische fouten kunnen opgespoord worden door het model enkele malen door te rekenen en de resultaten te vergelijken met die van het computerprogramma. Bij een model van deze omvang is dat nog goed mogelijk. Hierbij hebben we gebruik gemaakt van de tussenuitvoer en een simulatie over een klein aantal dagen uitgevoerd. Er werden geen logische fouten ontdekt.

Hoofdstuk 5 VALIDATIE AAN DE HAND VAN DE UITKOMSTEN

5.1 VALIDATIE VAN DE GEGENEREERDE VERDELINGEN

De uitvoerstaat van het computerprogramma geeft ons de volgende waarden (zie bijlage 2):

totaal aantal trekkingen	400
gemiddelde tussenaankomsttijd	119,58
standaarddeviatie tussenaankomsttijd	114,97
gemiddeld gewicht	29880,38
standaarddeviatie gewicht	3902,98
gemiddelde ontladingssnelheid	6990,00
standaarddeviatie ontladingssnelheid	1126,09

Deze waarden dienen op hun validiteit getoetst te worden.

TOETSING VAN DE AANKOMSTTIJD VAN DE VRACHTWAGENS

Van de negatief exponentiële verdeling wil men toetsen of de gemiddelde tussenaankomsttijd werkelijk 120 minuten is. Daarvoor stellen wij de volgende hypotheses:

$H_0 : \mu = 120$ en
$H_1 : \mu \neq 120$.

Verder is bekend dat $\alpha = 0{,}05$ (95% betrouwbaarheidsinterval) en $n = 400$. De gevonden gemiddelde aankomsttijd is 119,58 met een standaardafwijking van 114,97, zodat $\underline{\bar{x}} = 119{,}58$ en $\underline{s} = 114{,}97$.

De standaarddeviatie σ van de populatie is bekend, namelijk $\sigma = 120$. Als toetsingsgrootheid gebruiken we

$$\underline{z} = \frac{\underline{\bar{x}} - \mu}{\sigma/\sqrt{n}} ,$$

van de standaardnormale verdeling.

H_0 wordt aangenomen als $-1,96 < \underline{z} < 1,96$.
Bij ons kan H_0 worden aangenomen, want

$$\underline{z} = \frac{119,58 - 120,00}{120/\sqrt{400}} = 0,07$$

TOETSING VAN HET GEMIDDELD GEWICHT VAN DE LADINGEN

We willen toetsen of het gemiddeld gewicht van de ladingen 30.000 kg bedraagt. De hypothesen zijn dus:

$H_0 : \mu = 30.000$ en
$H_1 : \mu = 30.000$

Verder is bekend: $\alpha = 0,05$
$\underline{z} = \underline{x}$ $n = 400$
$\underline{s} = 3902,98$ en $\underline{x} = 29.880,39$

In dit geval is $\sigma = 4000$.
H_0 wordt aangenomen als $-1,96 < \underline{z} < 1,96$.
Nu is

$$\underline{z} = \frac{29880,39 - 300000,00}{4000/\sqrt{400}} \approx -0,60$$

H_0 wordt dus aanvaard.

TOETSING VAN DE ONTLADINGSSNELHEID

We willen toetsen of de gemiddelde ontladingssnelheid werkelijk 6900 kg/h bedraagt.

$H_0 : \mu = 6900$ en
$H_1 : \mu \neq 6900$.

$\alpha = 0,05$; $n = 400$; $\underline{x} = 6990$ en $\underline{s} = 1126,09$.
Ook hier geldt dat H_0 niet verworpen wordt als $-1,96 < \underline{z} < 1,96$.

$$\sigma = 1178$$

$$\underline{z} = \frac{6990 - 6900}{1178/\ 400} \approx 1,53,$$ zodat H_0 wordt aanvaard.

RESULTATENANALYSE

De computeruitvoer (bijlage 1) geeft ons de volgende resultaten:

RESULTATENANALYSE

De computeruitvoer (bijlage 1) geeft ons de volgende resultaten:

aantal heftrucks	gemiddelde kosten	standaard deviatie gem.kosten	gemiddelde overwerk minuten	standaard deviatie gem.overwerk
1	1744,84	67,03	656,37	23,41
2	1318,57	37,77	467,77	19,36
3	1324,97	27,49	376,12	15,89
4	1440,24	22,68	333,46	13,78
5	1599,19	21,15	318,68	13,14
6	1771,83	20,59	313,35	12,98
7	1950,97	20,52	313,04	12,97
8	2130,97	20,52	313,04	12,97

Het doel van ons onderzoek is het optimale aantal heftrucks te bepalen. Voor ons is echter niet geheel duidelijk of de opdrachtgever daarbij alleen geïnteresseerd is in minimalisatie van de kosten, of dat de mate van overwerk ook een rol kan spelen. Het blijkt namelijk dat de hoeveelheid te verrichten overwerk niet gering is en pas bij 7 heftrucks minimaal is (en dan nog gemiddeld 5 uur en 13 minuten per dag). Het overwerk is daarom een belangrijke kostenfactor (ca. 10% van de totale kosten). Voor ons onbekende gegevens uit bijvoorbeeld de CAO-afspraken zouden wel eens doorslaggevend kunnen zijn in dit probleem. Omdat dat soort gegevens voor ons niet bekend is, gaan we voorlopig toch maar uit van optimalisatie d.m.v. minimalisering van de totale kosten.

Uit de uitvoergegevens blijkt dat de gemiddelde kosten bij het gebruik van 2 heftrucks minimaal zijn. Het verschil in gemiddelde kosten tussen het gebruik van 2 of 3 heftrucks is klein. Een hypothesetoets omtrent dit verschil lijkt dan ook nodig. Als toetsingsgrootheid gebruiken we hiervoor:

$$\underline{z} = \frac{\underline{d} - \delta}{\sqrt{(\hat{\sigma}_1^2/n + \hat{\sigma}_2^2/n)}} \qquad \text{met} \qquad \delta = \mu_1 - \mu_2$$

Bij twee heftrucks is $\hat{\sigma}_2 = 755,33$ en $\underline{\bar{x}}_2 = 1318,57$

Bij drie heftrucks is $\hat{\sigma}_3 = 549,84$ en $\underline{\bar{x}}_3 = 1324,97$

$n_2 = n_3 = 400$, $H_0 : \mu_2 = \mu_3$, $H_1 : \mu_2 \neq \mu_3$.

Bij een 95% betrouwbaarheidsinterval betekent dit, dat H_0 niet verworpen wordt als: $-91,56 < \underline{d} < 91,56$.

$\underline{\bar{x}}_2 - \underline{\bar{x}}_3 = -6,4$, zodat H_0 waar is.

Het verschil tussen deze twee waarden is dus een normale afwijking en geen reden om te kiezen voor 2 heftrucks. Toetsen we met dezelfde toetsingsgrootheid het verschil tussen het gebruik van 2 of 4 en 3 of 4 heftrucks, dan worden de H -hypotheses wel verworpen. We moeten dus kiezen uit het gebruik van 2 of 3 heftrucks. Uit de uitvoergegevens blijkt dat de gemiddelde overwerkminuten per dag bij het gebruik van 3 heftrucks kleiner is. Ook hier toetsen we dit verschil.

2 heftrucks: $\bar{x}_2 = 467,77$ 3 heftrucks: $\bar{x}_3 = 376,12$

$\hat{\sigma}_2 = 387,74$ $\hat{\sigma}_3 = 317,85$

$n_2 = n_3 = 400$, $H_0 : \mu_2 = \mu_3$, $H_1 : \mu_2 \neq \mu_3$.

Bij een 95% betrouwbaarheidsinterval betekent dit, dat H_0 niet verworpen wordt, als $-49,09 < d < 49,09$.

$\bar{x}_2 - \bar{x}_3 = 91,65$, zodat H_0 verworpen wordt.

Het blijkt dus, dat het gebruik van 3 heftrucks theoretisch dezelfde gemiddelde kosten oplevert als het gebruik van 2 heftrucks. Echter, t.a.v. de gemiddelde overwerkminuten is het gebruik van 3 heftrucks aan te bevelen.

Hoofdstuk 6 CONCLUSIES

Uit de resultatenanalyse blijkt, dat bij de aangegeven doelstelling het gebruik van 3 heftrucks optimaal is.

Wij willen de opdrachtgever erop wijzen dat er ook nog een andere mogelijkheid tot minimalisering van de kosten en overwerktijd bestaat. De aankomsttijd van de vrachtauto's vindt binnen de normale werktijd plaats, het ontladen van deze auto's echter niet. Een vrachtwagen die 's middags om 20 voor 5 aankomt maakt gemiddeld 4 uur overwerk noodzakelijk. Door voor 2 chauffeurs de normale werktijd te wijzigen zou de overwerktijd verminderd kunnen worden. Zij zouden bijvoorbeeld om 9 uur kunnen beginnen. De kans op grote wachtrijen is dan gering, omdat de kans dat er het eerste uur van de dag 2 vrachtwagens arriveren nogal klein is.

Naast de hierboven geschetste mogelijkheid om de kosten te beperken, dient volgens ons ook nog door de opdrachtgever de mogelijkheid van het lossen van een wagen met 2 heftrucks bezien te worden.

BIJLAGE 1: Uitvoer computerprogramma

TOTAAL AANTAL TREKKINGEN	400
GEMIDDELDE TUSSENAANKOMSTTIJD	119.58
STANDAARDDEVIATIE TUSSENAANKOMSTTIJD	114.97
GEMIDDELDE GEWICHT	29880.38
STANDAARDDEVIATIE GEWICHT	3902.98
GEMIDDELDE ONTLADINGSSNELHEID	6990.00
STANDAARDDEVIATIE ONTLADINGSSNELHEID	1126.09

AANTAL HEFTRUCKS	GEMIDDELDE KOSTEN	STANDAARD DEVIATIE KOSTEN	GEMIDDELDE OVERWERK MINUTEN	STANDAARD DEVIATIE OVERWERK
1	1744.84	67.03	656.37	23.41
2	1318.57	37.77	467.77	19.36
3	1324.97	27.49	376.12	15.89
4	1440.24	22.68	333.46	13.78
5	1599.19	21.15	318.68	13.14
6	1771.83	20.59	313.35	12.98
7	1950.97	20.52	313.04	12.97
8	2130.97	20.52	313.04	12.97

BIJLAGE 2: Voorbeeld tussenuitvoer

DAGNUMMER 1
AANTAL VRACHTWAGENS VANDAAG 4
GEMIDDELDE TUSSENAANKOMSTTIJD VANDAAG 189.74

AANTAL HEFTRUCKS	KOSTEN VRACHT WAGENS	VASTE KOSTEN HEFTRUCKS	OVERWERK KOSTEN HEFTRUCKS	TOTAAL KOSTEN
1	739.88	180.00	359.46	1279.34
2	541.86	360.00	283.68	1185.54
3	507.28	540.00	237.57	1284.85
4	507.28	720.00	237.57	1464.85
5	507.28	900.00	237.57	1644.85
6	507.28	1080.00	237.57	1824.85
7	507.28	1260.00	237.57	2004.85
8	507.28	1440.00	237.57	2184.85

DAGNUMMER 2
AANTAL VRACHTWAGENS VANDAAG 3
GEMIDDELDE TUSSENAANKOMSTTIJD VANDAAG 183.19

AANTAL HEFTRUCKS	KOSTEN VRACHT WAGENS	VASTE KOSTEN HEFTRUCKS	OVERWERK KOSTEN HEFTRUCKS	TOTAAL KOSTEN
1	802.30	180.00	437.66	1419.97
2	523.84	360.00	292.94	1176.78
3	436.83	540.00	176.93	1153.76
4	436.83	720.00	176.93	1333.76
5	436.83	900.83	176.93	1513.76
6	436.83	1080.00	176.93	1693.78
7	436.83	1260.00	176.93	1873.76
8	436.83	1440.00	176.93	2053.76

DAGNUMMER 3
AANTAL VRACHTWAGENS VANDAAG 6
GEMIDDELDE TUSSENAANKOMSTTIJD VANDAAG 107.31

AANTAL HEFTRUCKS	KOSTEN VRACHT WAGENS	VASTE KOSTEN HEFTRUCKS	OVERWERK KOSTEN HEFTRUCKS	TOTAAL KOSTEN
1	2854.82	180.00	1115.87	4150.69
2	1629.62	360.00	898.69	2886.32
3	1214.86	540.00	695.11	2449.96

BESCHRIJVING VAN DE CASE ZIEKENHUISSIMULATIE*)

In een provincieplaats is een streekziekenhuis gepland. Op grond van de landelijke medische statistieken mag men verwachten dat uit het verzorgingsgebied een aanbod van 10.000 patiënten per jaar de polikliniek binnenkomt.

Hiervan is naar verwachting 60% poliklinisch te behandelen. De rest dient, al of niet als spoedpatiënt, opgenomen te worden. De verdeling over de afdelingen is als volgt:

afdeling	percentage	Gem. aantal ligdagen
Intern	15%	32
Chirurgie I	17%	20
Chirurgie II	13%	20
Gynaecologie	31%	12
Pediatrie	6%	24
K.N.O.	18%	3

Enkele opmerkingen

Het aantal spoedpatiënten bedraagt naar verwachting op alle afdelingen, behalve Chirurgie II, 40% van het totaal aantal patiënten. Deze afdelingen nemen in principe alleen op werkdagen patiënten op.

Chirurgie II neemt de gehele week patiënten op, dit zijn allen spoedpatiënten. Iedere ochtend wordt van iedere patiënt beslist of hij al dan niet moet blijven. Een vrijgekomen bed kan diezelfde dag weer door elk van de afdelingen in gebruik worden genomen.

Statistische gegevens zijn op aanvraag beschikbaar.

Opdracht:

- Doe voor dit ziekenhuis een voorstel betreffende het aantal bedden alsmede een te voeren opnamepolitiek.
- Stel een model op voor de bezetting van dit ziekenhuis gebruik makend van de verstrekte gegevens en uw voorstellen.
- Simuleer met dit model een periode van tenminste 10 weken normale bedrijfsvoering zodanig dat u aan het eind beschikt over de volgende gegevens:
  - De bezetting en de wachtrijlengte per dag;
  - De gemiddelde bezettingsgraad per afdeling voor werkdagen, zaterdagen en zondagen;
  - Aantal behandelde patiënten per afdeling en totaal;
  - Aantal doorgestuurde spoedpatiënten per afdeling en totaal;
  - Gemiddeld aantal ligdagen per afdeling per patiënt;
  - Gemiddeld aantal dagen op de wachtlijst per afdeling per patiënt.

*) Deze case is ter beschikking gesteld door Prof. Dr. A. Bosman

UITWERKING VAN DE CASE 'ZIEKENHUISSIMULATIE'*)

Hoofdstuk 1 PROBLEEMBESCHRIJVING

1.1 PROBLEEMANALYSE

De eerste fase zal een analyse van de probleembeschrijving inhouden. Wij zullen ons bezinnen op het vraagstuk en daarmee een aantal vragen formuleren. Het zal, gezien het feit dat we geen verdere informatie ter beschikking hebben dan uit de opgave naar voren komt, noodzakelijk zijn een aantal veronderstellingen te doen.

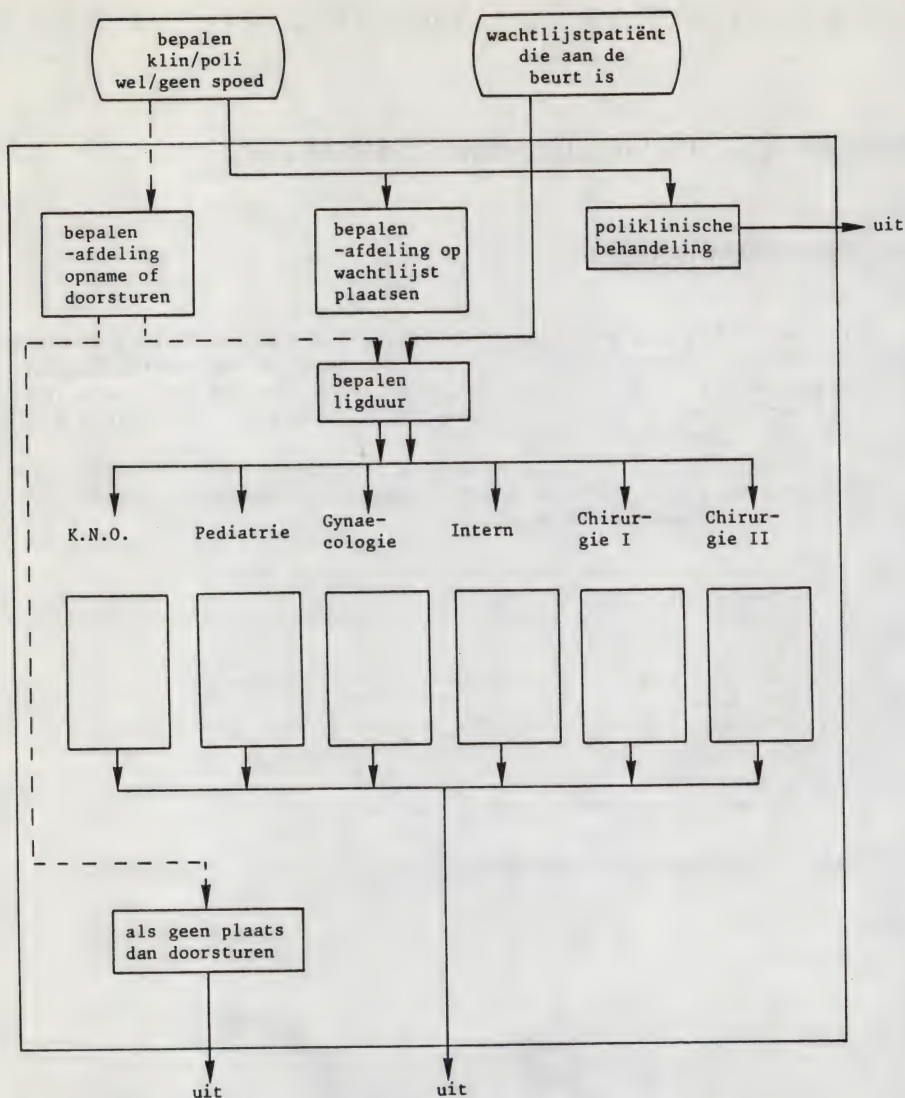
We starten met een globale schets van de ziekenhuisprocessen. Een beslissingstabel geeft weer welke acties moeten worden ondernomen, als een patiënt in het ziekenhuis binnenkomt.

klinische behandeling	J	J	J	N
spoedpatiënt	J	J	N	-
bed beschikbaar	J	N	-	-
patiënt opnemen	X			
patiënt doorsturen		X		
patiënt op wachtlijst			X	
patiënt behandelen				X

Vervolgens proberen we in een schets het probleem te visualiseren.



*) De uitwerking is gebaseerd op het verslag van het simulatiewerkstuk van N. van Driel en A. Rommens, studentes aan de Bedrijfskundige Informatica Opleiding aan de HEAO te Den Haag (1979).



Een schets van het ziekenhuissysteem

--------- = spoed
_________ = geen spoed

We zien snel in dat er binnen het systeem een aantal belangrijke componenten bestaat. Het zal nodig zijn om hierbij even stil te staan, zodat we in staat worden gesteld om het complexe systeem op te splitsen in meer overzichtelijke delen.

Op voorhand kunnen we zeker een onderscheid maken tussen de patiënten, de bedden en de opnamepolitiek.

1. *De patiënten*

De patiënten vormen de kern van het ziekenhuis, zij zijn de spil waar alles om draait. Het patiëntenaanbod bepaalt hoeveel bedden er nog zijn om alle patiënten te kunnen behandelen. De bepalende factoren daarbij zijn het aantal patiënten dat een behandeling nodig heeft èn de gemiddelde tijd gedurende welke een patiënt een bed bezet houdt. Tevens is de reden waarom de patiënt behandeld moet worden bepalend t.a.v. de vraag of die behandeling direct plaats moet vinden. Is dat niet nodig, dan kan eventueel bekeken worden in hoeverre hij voorrang heeft op andere, reeds wachtende, patiënten. Bij het opbouwen van het simulatiemodel is het vaststellen van de omvang en de samenstelling van het patiëntenaanbod dus van groot belang.

2. *De bedden*

Bedden fungeren als een belangrijk hulpmiddel bij de behandeling van de patiënten. Als er geen bed beschikbaar is, is er meestal geen behandeling mogelijk en de patiënt moet wachten of, als dat niet kan, worden doorgestuurd naar een ander ziekenhuis.

Met betrekking tot de plaatsing van de bedden kunnen verschillende mogelijkheden worden onderscheiden. Eén mogelijkheid is om elke afdeling een bepaald aantal bedden toe te wijzen en die verdeling altijd te handhaven, d.w.z. de bedden van een bepaalde afdeling worden nooit door andere afdelingen gebruikt. Een andere mogelijkheid is die, waarbij wèl onderling bedden kunnen worden uitgewisseld. Bij beide mogelijkheden bestaat verder de keuze tussen het al dan niet reserveren van bedden voor eventuele spoedpatiënten. Deze reservering kan per afdeling gebeuren, maar ook een algemene reservevoorraad behoort tot de mogelijkheden.

3. *De opnamepolitiek*

Een belangrijke factor met betrekking tot de gevoerde opnamepolitiek is de keuze t.a.v. het afsprakenbeleid. De eerste mogelijkheid daarbij is om niet te werken met afspraaktermijnen, d.w.z. een patiënt die een behandeling moet ondergaan wordt als er een bed is altijd opgenomen. Eventueel kunnen wel bedden worden vrijgelaten voor spoedpatiënten.

Het voordeel van deze methode is dat er geen uitgebreide afsprakenprocedure nodig is, behalve voor de patiënten voor wie niet direct een bed beschikbaar is. Verder hoeft de patiënt niet langer met zijn kwaal te blijven rondlopen. Het nadeel van deze methode is dat de opname de patiënt kan overvallen, d.w.z. hij heeft geen gelegenheid bepaalde zaken, zoals zorgen voor hulp in het gezin, het inwerken van een tijdelijke vervanger op het werk, van tevoren te regelen. Ook moet men van de kant van het ziekenhuis maar afwachten hoe de bezetting e.d. in een bepaalde periode zullen zijn. Dat kan minder goed geregeld worden, dan wanneer gebruik wordt gemaakt van de tweede mogelijkheid, namelijk wèl werken met afspraaktermijnen.

Het nadeel van deze tweede methode is de benodigde afspraken-procedure. Naast het reeds genoemde voordeel voor het ziekenhuis, geeft deze methode de patiënt bovendien de mogelijkheid om het een en ander te regelen voordat hij wordt opgenomen.

Een andere belangrijke factor is het al dan niet werken met prioriteiten, d.w.z. de ene patiënt voorrang verlenen boven een andere.

Verder moet er bepaald worden hoe het opnamebeleid in het weekend zal zijn. Ook de aanpak t.a.v. het in 2 genoemde gebruik van de bedden behoort in feite tot de opnamepolitiek.

Na het voorbereidende denkwerk is het duidelijk geworden dat het ziekenhuisprobleem tot de klasse 'wachtrijproblemen' behoort. Hoe die rijen georganiseerd moeten worden is van later zorg. Wel kunnen we in een schets de problematiek overzichtelijk weergeven.

Wachtrijen	Afdelingen	Aankomst-verdeling	ligduur in dagen
.... 40% spoed →	K.N.O.	18%	3
.... 40% spoed →	Pediatrie	6%	24
.... 40% spoed →	Gynaecologie	31%	12
.... 40% spoed →	Intern	15%	32
.... 40% spoed →	Chirurgie I	17%	20
100% spoed →	Chirurgie II	13% (Over 7 dagen)	20

1.2 DE DOELSTELLING VAN DE SIMULATIE

Het gestelde probleem kan o.a. met behulp van simulatie worden opgelost. De doelstelling van de ziekenhuissimulatie kan daarom als volgt worden geformuleerd.

Het verkrijgen van een aantal gegevens met behulp waarvan het benodigde aantal bedden en de te voeren opnamepolitiek door de ziekenhuisdirectie kan worden vastgesteld.

Deze doelstelling kan worden bereikt m.b.v. een simulatiemodel een aantal weken bedrijfsvoering te simuleren, zodanig dat men daarna over de gewenste gegevens kan beschikken. Deze gegevens zijn de volgende:

- de bezetting en de wachtrijlengte per dag;
- de gemiddelde bezettingsgraad per afdeling voor werkdagen, zaterdagen en zondagen;
- aantal behandelde patiënten per afdeling en totaal;
- aantal doorgestuurde spoedpatiënten per afdeling en totaal;
- gemiddeld aantal ligdagen per afdeling per patiënt;
- gemiddeld aantal dagen op de wachtlijst per afdeling per patiënt.

De ziekenhuisdirectie heeft t.a.v. deze gegevens bepaalde eisen. Dat wil zeggen, ze heeft bepaalde criteria m.b.t. de waarden die deze gegevens (variabelen) mogen hebben. Zij zal nu op grond van de resultaten die het simulatiemodel oplevert dàt aantal bedden en díe opnamepolitiek kiezen, die gegeven haar doelstellingen het meest bevredigend zijn.

Hoofdstuk 2 SYSTEEMDEFINITIE

2.1 DE COMPONENTEN IN HET ZIEKENHUISSYSTEEM

Alvorens te komen tot een specificatie van het model is het goed het model eerst in een groter verband te bekijken. Elk model is een afbeelding van een reëel systeem; in dit geval is dat systeem een ziekenhuis. Binnen het systeem ziekenhuis kunnen we vele componenten onderscheiden, de belangrijkste daarvan zijn de volgende:

- ziekenhuisdirectie
- het gebouw
- de afdelingen
- de polikliniek
- het laboratorium
- de artsen
- het verplegend personeel
- het dienstrooster
- de bedden
- het instrumentarium
- de medicijnen
- de patiënten
- de ligduur van een patiënt
- het opnamebeleid
- de wachtlijst
- bezoektijden regeling

Het is natuurlijk ondoenlijk al deze componenten in het model op te nemen en dat is gezien de doelstelling van het simulatieproject ook niet noodzakelijk. De primaire doelstelling van het hele ziekenhuisgebeuren is de geneeskundige behandeling van iedereen die daar behoefte aan heeft zo goed mogelijk te laten verlopen. Dat is iets heel anders dan de in Hoofdstuk 1 gespecificeerde doelstelling van het simulatieproject.

2.2 GRENZEN VAN HET SYSTEEM

Bij het definiëren van het te bestuderen systeem kunnen twee soorten grenzen worden onderscheiden:

1. de grenzen die het ziekenhuissysteem scheiden van de rest van de samenleving;
2. de grenzen die het subsysteem, dat onderwerp van studie is scheiden van de rest van het ziekenhuissysteem.

Ad. 1. Door deze systeemgrenzen heen loopt een aantal relaties die het ziekenhuissysteem heeft met de rest van de maatschappij. Zo zijn er de patiënten die via deze grenzen het ziekenhuis binnenkomen. Er is de overheid die bepaalde maatregelen kan treffen, waarvan de invloed tot in het ziekenhuis voelbaar kan zijn. Verder zijn er de andere ziekenhuizen in de omgeving, die bijvoorbeeld gebruikt kunnen worden voor de opvang van spoedpatiënten voor wie geen plaats is.
Al deze elementen zijn voor het ziekenhuis deel van de buitenwereld. Vaak elementen waaraan niet veel veranderd kan worden en die dus als een gegeven beschouwd moeten worden. Zo zal bijvoorbeeld, uitgezonderd van de werking die van een goede voorlichting uit kan gaan, weinig invloed uitgeoefend kunnen worden op het aantal mensen dat een been breekt.

Ad. 2. Op basis van de geformuleerde doelstelling kan een bepaald aspect van het ziekenhuissysteem als onderwerp van studie worden aangeduid. Dat aspect kan worden aangemerkt als: 'de regeling van de behandeling van de patiënten', d.w.z. het regelen van opnames, van het gebruik van de bedden e.d. Het personeelsaspect bijvoorbeeld blijft hier buiten beschouwing.

In 2.1 zijn de belangrijkste componenten van het ziekenhuissysteem genoemd. Het zal duidelijk zijn dat er uitgaande van het onder Ad. 2. besprokene slechts een aantal van deze componenten in het model hoeft te worden opgenomen.
Welke componenten dat zijn zal in hoofdstuk 3 worden aangegeven. Een eerste afbakening van het simulatiemodel wordt gevormd door de in de volgende paragraaf opgenomen veronderstellingen.

2.3 EEN AANTAL UITGANGSPUNTEN

Bij de opstelling van het simulatiemodel zal worden uitgegaan van een aantal veronderstellingen. De belangrijkste daarvan zijn:

1. Bij aankomst van een patiënt in het ziekenhuis is bekend of het een poliklinische of klinische behandeling betreft. Ook is bekend of er sprake is van een spoedpatiënt of niet. Tenslotte is ook de afdeling waar de patiënt moet worden opgenomen direct te bepalen.
2. Op de poliklinische behandeling wordt in het model verder niet ingegaan.
3. Als er sprake is van een spoedpatiënt dan wordt deze altijd opgenomen met als enige voorwaarde dat er een bed vrij is; is dit niet het geval dan wordt de patiënt doorgestuurd naar een ander ziekenhuis.
4. Er wordt afgezien van patiënten die gedurende één verblijf in het ziekenhuis van afdeling veranderen. In de praktijk zal dat percentage ook niet zo groot zijn. (In elk geval niet voor de in het model onderscheiden afdelingen.)

5. Met patiënten die gedurende hun verblijf in het ziekenhuis overlijden wordt niet expliciet rekening gehouden. Deze gebeurtenis wordt in het geheel betrokken door dit als een voortijdig einde van de ligduur te beschouwen. Aangenomen wordt dat dergelijke situaties in de gemiddelde ligduur van de verschillende afdelingen zijn verwerkt.
6. Bij de plaatsing van de bedden wordt ervan uitgegaan dat er fysiek geen beperking is t.a.v. het aantal bedden dat op een bepaalde afdeling mogelijk is. Dit lijkt een redelijke veronderstelling, omdat er sprake is van een nog te bouwen ziekenhuis.
7. Bij de bouw van het model wordt aangenomen, dat het betrekking heeft op de normale bedrijfsvoering; er wordt geabstraheerd van duidelijk afwijkende situaties als:

   - gijzelingsacties
   - grote rampen (vliegtuig/treinongelukken)



Hoofdstuk 3 MODELFORMULERING

3.1 DE DOELSTELLING VAN HET MODEL

Zoals reeds bij de probleem- en doelstellingbeschrijving in hoofdstuk 1 is aangegeven, is de ziekenhuissimulatie gericht op het verkrijgen van een aantal gegevens aan de hand waarvan de ziekenhuisdirectie het benodigde aantal bedden en de te voeren opnamepolitiek kan vaststellen. Het doel van

het model is dus het vergelijken van een aantal alternatieven, het model moet daartoe als resultaat van elk gesimuleerd alternatief een aantal gegevens opleveren. Deze alternatieven verschillen voor wat betreft het aantal bedden en de opnamepolitiek. Dit vergelijkende karakter van het model impliceert dat vooral verschillen tussen de uitvoervariabelen bij de verschillende alternatieven van belang zijn en dat het model niet primair is gericht op het zo nauwkeurig mogelijk afbeelden van de werkelijkheid.

3.2 DE MODELCOMPONENTEN

Van de in 2.1 genoemde componenten zijn de volgende voor ons simulatiemodel relevant:

- de afdelingen
- de polikliniek
- de bedden
- de patiënten
- de ligduur van de patiënten
- het opnamebeleid
- de wachtlijst (patiënten met een afspraak voor behandeling)

Opgemerkt kan worden dat de tweede component, de polikliniek, een wat bijzondere afdeling is. Verder is de ligduur van de patiënt een bepaald gegeven dat bij die patiënt hoort. De eerste drie componenten zijn vaste componenten van het ziekenhuismodel, daarentegen is de vierde component, de patiënten, een tijdelijke. De componenten ligduur, opnamebeeld en wachtlijst zijn in dit opzicht wat moeilijk in te delen.

3.3 DE PARAMETERS EN VARIABELEN

3.3.1 De parameters

De belangrijkste parameters die in het model onderscheiden kunnen worden zijn:

1. voor elke afdeling: het gemiddelde en de standaardafwijking van de verdeling van de ligduur van de patiënten;
2. de percentages voor de verdeling van het patiëntenaanbod over de verschillende afdelingen en de polikliniek;
3. de percentages voor de verdeling van het patiëntenaanbod in spoed- en niet-spoedpatiënten.

3.3.2 De variabelen

Bij elk model zijn globaal altijd twee soorten variabelen te onderscheiden. Enerzijds zijn er de al dan niet beheersbare variabelen die als invoer voor het model dienen. Anderzijds zijn er de variabelen die als uitvoer het model verlaten, de endogene variabelen. Naast deze twee groepen zijn er meestal nog een aantal toestandsvariabelen aanwezig. Voor het ziekenhuissimulatiemodel kunnen de volgende variabelen worden onderscheiden.

Invoervariabelen

1. *Exogene variabelen*
   1. het patiëntenaanbod;
   2. de afdeling waar een patiënt moet worden opgenomen;
   3. de ligduur van een patiënt;
   4. de mate van urgent zijn van de opname van een patiënt.

Ad.1.1 In de probleembeschrijving in hoofdstuk 1 is een verwacht patiëntenaanbod van in totaal 10.000 patiënten per jaar opgenomen.

Ad.1.2 Ook over deze variabele zijn gegevens uit de probleembeschrijving beschikbaar, de volgende afdelingen worden onderscheiden:
1. de polikliniek;
2. de klinische afdelingen:
   (a) Intern
   (b) Chirurgie I
   (c) Chirurgie II
   (d) Gynaecologie
   (e) Pediatrie
   (f) K.N.O.

2. *Instrumentele variabelen*
   1. het aantal ziekenhuisbedden;
   2. de opnamepolitiek;

Ad.2.2 Het is mogelijk om deze variabele nader te detailleren door niet uit te gaan van een universele opnamepolitiek voor alle afdelingen, maar deze te variëren voor de verschillende afdelingen. Verder kan voor de niet-spoedpatiënten bij het op de wachtlijst plaatsen gewerkt worden met prioriteiten en ook die kunnen voor elke afdeling anders zijn.

Als instrumentele variabelen zouden eventueel ook denkbaar kunnen zijn:

- een maximum toegestaan percentage of een absoluut aantal toegestane spoedpatiënten dat doorgestuurd moet worden;
- een bepaalde minimumeis t.a.v. de bezettingsgraad.

Dit zijn criteria die de ziekenhuisdirectie kan toepassen op de simulatieresultaten. Ze worden niet als zodanig in het model opgenomen.

Endogene variabelen

Deze komen sterk overeen met de gegevens die het model (per alternatief) moet opleveren.

1. de bezettingsgraad van de bedden, per afdeling;
2. de wachtrijlengte, per afdeling;
3. het aantal behandelde patiënten, per afdeling;
4. het aantal doorgestuurde patiënten, per afdeling;
5. gemiddeld aantal ligdagen van een patiënt, per afdeling;
6. gemiddeld aantal dagen dat een patiënt op de wachtlijst staat, per afdeling.

Toestandsvariabelen

1. de dag in de week;
2. de periode van de dag
3. het tijdstip op een dag;
4. de waarde van de verschillende rijen.

Ad.1 Er moet verschil gemaakt worden tussen de werkdagen (maandag t/m vrijdag) en de weekenddagen, zaterdag en zondag o.a. voor de opnamepolitiek.

Ad.2 Er bestaat een verschil tussen de periode van de dag waarin alleen maar spoedpatiënten aankomen ('s nachts) en de periode waarin ook patiënten aankomen die niet meteen opgenomen behoeven te worden (overdag).

Ad.3 Een belangrijk tijdstip van de dag, is een bepaalde tijd 's ochtends, bijvoorbeeld om 8.00 uur, wanneer voor elke patiënt bekeken wordt of hij het ziekenhuis mag verlaten.

Ad.4 Bijvoorbeeld het aantal vrije en bezette bedden per dag, alsmede het aantal patiënten dat op opname wacht.

3.4 DE RELATIES IN HET MODEL

Er bestaan in dit dynamische, stochastische model vele relaties tussen de verschillende invoer- en uitvoervariabelen.

Het doel van het model is het opleveren van gegevens ter vergelijking van alternatieven en niet het nauwkeurig kwalificeren van de relaties.

Eerst wordt in een relatietabel in kaart gebracht welke oorzaak-gevolg relaties er tussen de meeste relevante variabelen bestaan.

De aanduiding positief (pos) wil zeggen dat een toename van de betreffende invoervariabele de endogene variabele doet toenemen (bij constant gehouden overige invoervariabelen). Is de aanduiding negatief (neg) dan wil dat zeggen dat een toename van de invoervariabele de endogene variabele doet afnemen.

	bezettingsgraad	aantal behandelde patiënten	aantal doorgestuurde spoed-patiënten	aantal dagen op wachtlijst
aantal bedden	negatief	positief	negatief	negatief
opnamebeleid		geen eenduidige relatie		
patiëntenaanbod	positief	positief	positief	positief

Het zal duidelijk zijn dat de aanduiding positief en negatief slechts zeer globale relaties aanduidt.

Deze tabel kan dienen om de gestelde hypothetische verbanden te toetsen aan de resultaten van de simulatie-experimenten.

Omdat de relaties niet in de vorm van functies worden geformuleerd, dienen we ze op een andere wijze van inhoud te voorzien.

Dit gebeurt in de vorm van een processchema. Dit blijft hier nog globaal; tijdens de programmeringsfase wordt het proces gedetailleerd beschreven.

Per dag kunnen we de volgende gebeurtenissen onderscheiden:

1. Patiënten komen aan:
   - poliklinisch = behandelen
   - klinisch = - spoed: zo mogelijk opnemen
     - niet-spoed: afspraak maken

Bij het verwerken van het aanbod van patiënten moeten de verschillende opnamepolitieken verwerkt worden.

2. Patiënten worden opgenomen:
   - een bed wordt bezet
   - de ontslagdag wordt bepaald.

3. Patiënten worden ontslagen:
   - er komt een bed vrij.

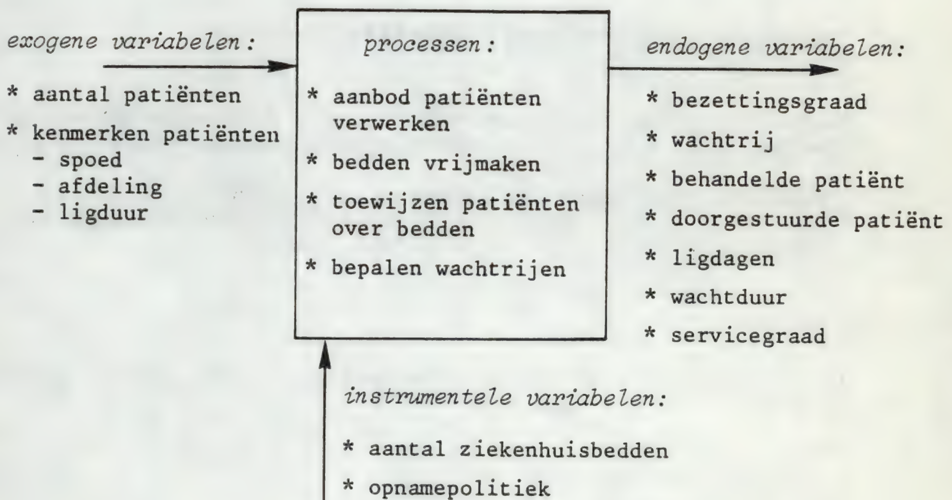
Per dag moeten we een aantal gegevens bijwerken die nodig zijn om de gewenste uitvoer te krijgen.

N.B. Het ligt voor de hand dat tijdens de verwerking éérst de ontslagen patiënten worden verwerkt, zodat vrijgekomen bedden dezelfde dag nog kunnen worden bezet.

Globaal proces, per dag en per afdeling

Ontslaan van de genezen patiënten.
Zolang de dag niet beëindigd is:
- aankomst van een patiënt
  bepaal spoed
  bepaal ligduur
  bepaal wachtrij
  alloceer een bed
  neem spoed op
  als er geen bedden meer zijn dan
  - doorsturen
  - andere afdeling
  - op de gang zetten

We kunnen het model samenvattend in een schema onderbrengen:



Hoofdstuk 4 EXPERIMENTEEL ONTWERP

4.1 WAT HOUDT HET EXPERIMENTEEL ONTWERP IN?

Voordat we het geformuleerde model omzetten in een model, dat op de computer kan draaien zal eerst een meer gedetailleerd experiment moeten worden opgezet, opdat de gewenste experimentele data op de meest efficiënte manier ter beschikking komen. Bij de formulering van het model is al aangegeven (3.3) welke parameters en variabelen in het model onderscheiden worden. In dit hoofdstuk worden deze variabelen uitgedrukt door de termen 'factoren' en 'responses', waarbij de 'factoren' corresponderen met de genoemde invoervariabelen en de 'responses' met de genoemde endogene variabelen.

Eén van de eerste beslissingen, die we moeten nemen bij het bepalen van de wijze waarop we het model zullen draaien en de resultaten zullen analyseren, is het bepalen van het 'datum point'. Bij het bepalen van wat bij de simulatie één 'datum point' is kunnen verschillende mogelijkheden worden onderscheiden. In dit geval is het waarschijnlijk het beste om als 'datum point' een bepaald gedeelte van de gesimuleerde tijd in elke run te kiezen, bijvoorbeeld een dag, een week of een aantal weken. (Eén run is minimaal 10 weken.)

Op basis van het doel van het simulatiemodel kan gesteld worden dat de in dit geval vereiste statistische analyse zal bestaan uit het vergelijken van gemiddelden en varianties van alternatieven. De belangrijkste problemen bij het opzetten van het experiment zijn daarom dan ook: het bepalen van de steekproefgrootte, de startcondities en het onderzoeken van de aanwezigheid van autocorrelatie.

4.2 CRITERIA VOOR HET EXPERIMENTEEL ONTWERP

Aangezien het uiteindelijke ontwerp in sterke mate wordt bepaald door de ontwerpcriteria, zal nader worden ingegaan op de volgende vragen:

- Welke 'factoren' worden in het model gevarieerd?
- Wat valt er op te merken over de 'waarden' van de verschillende 'factoren'?

4.2.1 Factoren

Bij het bepalen van de ontwerpcriteria spelen de factoren een belangrijke rol; de factoren van het model zijn:

- het aantal bedden;
- de opnamepolitiek;
- het patiëntenaanbod;
- het gedeelte van de week;
- de ligduur van de patiënten op de verschillende afdelingen;
- de percentages voor de verdeling over de afdelingen;
- de percentages voor de verdeling in spoed- en niet-spoedpatiënten.

Vier van deze factoren worden in het model als constante meegenomen, omdat we voor deze factoren slechts één waarde onderscheiden, namelijk een bepaalde verdeling met één vast gemiddelde. Het zijn factoren, die door de systeemomgeving worden bepaald en als zodanig dus niet beheersbaar zijn. Bij het opzetten van het experiment gaan we ervan uit dat deze factoren een bepaalde verdeling hebben met een constant gemiddelde.

De bedoelde factoren zijn de volgende:

- ligduur;
- patiëntenaanbod;
- afdelingsverdeling;
- spoed-/niet-spoedverdeling.

De ligduur

De ligduur van een patiënt kan beschouwd worden als een soort bedieningstijd. In de wachttijdtheorie is gebleken dat bedieningstijden vaak negatief exponentieel verdeeld zijn. Daarom is voor de ligduur een negatief exponentiële verdeling gekozen met de volgende gemiddelden:

Afdeling	μ
Intern	32
Chirurgie I	20
Chirurgie II	20
Gynaecologie	12
Pediatrie	24
K.N.O.	3

Voor een verdere validatie van deze keuze wordt verwezen naar bijlage 1, waarin de χ^2-toets wordt toegepast op de ligduur van 200 chirurgiepatiënten.

Het patiëntenaanbod

Van het patiëntenaanbod is alleen gegeven dat men per jaar 10.000 patiënten verwacht. Als verdeling hiervoor wordt de Poissonverdeling gekozen, omdat uit dezelfde wachttijdtheorie is gebleken dat vele aankomstpatronen de vorm hebben van een Poissonverdeling. Voor een verdere validatie wordt verwezen naar de χ^2-toets toegepast op het aanbod voor de afdeling gynaecologie voor 200 werkdagen, te vinden in bijlage 2.

Verdeling van de patiënten

Voor de verdeling van de patiënten over de afdelingen en voor de verdeling van de patiënten in spoed- en niet-spoedpatiënten worden empirische verdelingen gebruikt, die in het programma ingevoerd moeten worden. Daarbij wordt gewerkt met de percentages zoals deze in de probleembeschrijving in hoofdstuk 1 zijn aangegeven. We hebben dus gezien dat een aantal factoren wel aan variatie onderhevig is. Dit betreft het aantal bedden, de opnamepolitiek en het gedeelte van de week.

4.2.2 Waarden van de factoren

Zoals in de voorafgaande paragraaf is aangegeven, zal slechts gekeken worden naar de waarden van drie factoren. T.a.v. het al dan niet kwantificeerbaar/beheersbaar zijn van de overgebleven factoren (en dus ook van de waarden van deze factoren) kan worden opgemerkt:

- aantal bedden = kwantificeerbaar, beheersbaar
- opnamepolitiek = kwalitatieve factor, beheersbaar
- gedeelte van de week = niet beheersbare factor.

De waarden van het aantal bedden en de opnamepolitiek kunnen vrij bepaald worden. De week is echter verdeeld in een vast aantal dagen en is dus als zodanig niet beheersbaar.

4.3 HET STRUCTUREEL MODEL VAN HET EXPERIMENTEEL ONTWERP

In het structureel model wordt het aantal factoren en het aantal waarden per factor beschreven, zoals de onderzoeker dat o.b.v. zijn doelstellingen het liefst zou hebben. Zoals in de vorige paragrafen is aangegeven, zijn er drie factoren overgebleven die in dit stadium voor het model van belang zijn. Deze factoren zijn tevens die variabelen die bij het bepalen van de doelstelling van het model naar voren kwamen. Van elke factor zullen nu de waarden van die factor worden beschreven.

Aantal bedden

In principe is het aantal waarden van deze factor 'oneindig groot' (wel uitsluitend gehele waarden). Er zal een schatting gemaakt moeten worden van het aantal benodigde bedden. In de buurt van deze schatting kan dan nog een beperkt aantal waarden gekozen worden.

Opnamepolitiek

Ook voor deze factor is een groot aantal waarden mogelijk. Dit komt vooral doordat het opnamebeleid in feite uit een aantal facetten bestaat. De mogelijke waarden zullen dan ook in een aantal groepen worden beschreven. Eén opnamepolitiek bestaat dan eigenlijk uit een combinatie van 'subwaarden', uit elke groep steeds één.

1. Afsprakenbeleid
   - geen afspraaktermijnen;
   - de afspraaktermijn van een patiënt is alleen afhankelijk van de afdeling waar hij moet worden opgenomen;
   - de afspraaktermijn is behalve van de afdeling ook afhankelijk van de prioriteit die een patiënt krijgt toegewezen;
   - de afspraaktermijn is afhankelijk van het aantal vrije bedden op de afdeling en van de wachtrij van de afdeling;
   - de afspraaktermijn is slechts afhankelijk van de prioriteit van de patiënt;
   - vaste afspraaktermijn voor het hele ziekenhuis.

2. **Beleid in het weekend**

- de gehele week spoed èn niet-spoedpatiënten opnemen;
- in het weekend uitgezonderd de spoedpatiënten geen patiënten opnemen;
- op alle dagen ontslaan;
- in het weekend (zaterdag en zondag) niet ontslaan;
- mensen die te lang op de wachtlijst staan gaan beschouwen als spoedpatiënten en ze dus eventueel zelfs in het weekend opnemen.

3. **Beddengebruik**

- elke afdeling gebruikt uitsluitend zijn eigen bedden;
- er wordt op elke afdeling een aantal bedden gereserveerd voor eventuele spoedpatiënten;
- er is een algemene reservebeddenvoorraad;
- de bedden op een afdeling worden verdeeld in bedden, die uitsluitend gebruikt mogen worden door spoedpatiënten en een aantal dat uitsluitend gebruikt mag worden door niet-spoedpatiënten;
- als er op een bepaalde afdeling geen bed beschikbaar is, dan kan voor een spoedpatiënt van deze afdeling een vrij bed van één van de andere afdelingen gebruikt worden.

Gedeelte van de week

Het aantal waarden van deze factor is vrij beperkt. De handigste indeling voor het model is de volgende:

- maandag t/m vrijdag
- zaterdag
- zondag

Eventueel zijn andere combinaties mogelijk, dit is vooral afhankelijk van de gebruikte opnamepolitiek; deze keuze komt echter pas later aan de orde.

Aangezien het experiment veel te groot zou worden als we alle eerdergenoemde waarden in het model zouden opnemen zal in het functioneel model worden aangegeven welke waarden inderdaad in het model worden opgenomen. T.a.v. de opnamepolitiek kan reeds worden vermeld, dat voor de groepen 'Beleid in het weekend' en 'Beddengebruik', één bepaalde waarde wordt gekozen en dat dus in principe alleen het afsprakenbeleid gevarieerd wordt. Naderhand kan dan eventueel een soort gevoeligheidsanalyse worden toegepast, door voor de andere twee groepen een aantal andere waarden uit te testen.

4.4 HET FUNCTIONEEL MODEL VAN HET EXPERIMENTEEL ONTWERP

In het functioneel model wordt beschreven welke factoren en welke waarden uiteindelijk in het model worden opgenomen.

Alle factoren uit het structurele model zullen in het functionele model worden opgenomen, omdat ze geen van allen gemist kunnen worden om de doelstelling van het model te bereiken. Per factor zal nu het aantal relevante waarden worden aangegeven.

Aantal bedden

Voor het aantal bedden zal in principe worden uitgegaan van: 160 en 180 bedden.

Dit is gebaseerd op een beddennorm zoals die voor de beoordeling van plannen in de ziekenhuiswereld wordt gehanteerd (zie bijlage 3). De algemeen geldende norm voor het aantal ziekenhuisbedden van een algemeen ziekenhuis is 4 pro mille, d.w.z. 4 bedden per 1.000 inwoners.

Een kleine omrekening levert dan een beddenaantal op van 160, namelijk:

- klinisch patiëntenaanbod = 4.000 per jaar, als de opnamecoëfficiënt 10% is, dan is dat dus 10% van het aantal inwoners;
- het aantal inwoners is dan: 10 × 4.000 = 40.000;
- bij een beddennorm van 4 pro mille levert dat dus een beddenaantal op van 0,004 × 40.000 = 160.

Bij deze berekening wordt er wel van uitgegaan, dat de hierboven gebruikte gegevens gebaseerd zijn op dezelfde medische statistieken die in de probleembeschrijving zijn aangeduid.

Gedeelte van de week

Voor deze factor worden dezelfde waarden als in het structurele model aangehouden, omdat deze indeling nodig is om de gewenste gegevens op te kunnen leveren (m.b.t. de bezettingsgraden).

Opnamepolitiek

Zoals in paragraaf 4.3 bij het structurele model is opgemerkt zal bij de opnamepolitiek van een aantal vaste 'subwaarden' worden uitgegaan. Dat zijn de volgende:

1. op zaterdag en zondag worden uitsluitend spoedpatiënten opgenomen, er worden geen afspraken gemaakt voor die dagen;
2. op zondag worden geen patiënten ontslagen, mensen die zondags ontslagen zouden moeten worden, moeten wachten tot maandag;
3. voor de niet-spoedpatiënten wordt steeds gewerkt met afspraken;
4. op elke afdeling wordt een aantal bedden gereserveerd voor spoedpatiënten: als er een patiënt komt die een afspraak heeft dan wordt hij alleen opgenomen als er voor eventuele spoedpatiënten genoeg bedden overblijven;
5. elke afdeling gebruikt uitsluitend zijn eigen bedden, er wordt niet geschoven tussen de afdelingen.

Daarnaast worden de volgende waarden van het 'Afsprakenbeleid' onderscheiden:

1. De afspraaktermijn van de patiënt is alléén afhankelijk van de afdeling waar hij moet worden opgenomen. Elke afdeling heeft een eigen afspraaktermijn.
2. De afspraaktermijn is behalve van de afdeling ook afhankelijk van de prioriteit die een patiënt krijgt toegewezen. Deze prioriteit is afhankelijk van de aard en de ernst van hun kwaal. Deze prioriteit wordt voor het model als een gegeven beschouwd. We zullen bij deze werkwijze drie prioriteiten onderscheiden.

3. De afspraaktermijn is afhankelijk van de afdeling, van het aantal vrije bedden en van de wachtlijst van de afdeling. Als een patiënt zich aanmeldt dan wordt gekeken of de afdeling op dat moment erg vol is en hoe lang de wachtlijst is. Aan de hand van deze gegevens wordt met de patiënt een afspraak gemaakt.

Bij het opzetten van het functionele model wordt ook een uitspraak gedaan over het aantal herhalingen. Het aantal herhalingen is o.a. afhankelijk van wat we als 'datum point' nemen. Voor 'datum point' komen bijvoorbeeld in aanmerking:

- een periode van 10 weken, 10 weken = 1 herhaling
- een periode van 1 week, 10 weken = 10 herhalingen
- een periode van 1 dag, 10 weken = 70 herhalingen

Als we deze laatste mogelijkheden als 'datum point' nemen, lopen we wel het gevaar dat er autocorrelatie in de steekproefuitkomsten aanwezig is. Deze autocorrelatie is het gevolg van het karakter van het ziekenhuisgebeuren. Er is dan een correctie nodig.
Onze kennis van autocorrelatie is echter te beperkt om het mogelijk te maken de steekproefuitkomsten te corrigeren. Daarom is gekozen voor een 'datum point' van tien weken (zoals de opdracht luidde) en slechts één herhaling van deze tien weken. Dit laatste vanwege kosten- en tijdoverwegingen.

Opmerking

Het laatste argument is niet steekhoudend voor een 'reële' situatie, waarin men via de simulatie-experimenten een beleidskeuze zou moeten maken. Dan dient men terdege te onderzoeken hoe lang de simulatietijd moet zijn (zeker meer dan 10 weken) en hoeveel herhalingen nodig zijn om aan een te voren vastgestelde gewenste betrouwbaarheid te kunnen voldoen.

4.5 HET UITEINDELIJKE EXPERIMENTEEL MODEL

Bij een experimenteel model worden de factoren en waarden als een geheel bekeken. De combinatiemogelijkheden van de verschillende factoren moeten nu onderzocht worden, het zogenaamde 'factorial design'.
De verschillende combinaties die mogelijk zijn worden in het onderstaande schema weergegeven.

	GEDEELTE VAN DE WEEK	AANTAL BEDDEN 1			AANTAL BEDDEN 2		
		OPNAMEPOLITIEK 1	2	3	OPNAMEPOLITIEK 1	2	3
1	maandag t/m vrijdag	x	x	x	x	x	x
2	zaterdag	x	x	x	x	x	x
3	zondag	x	x	x	x	x	x

Er zijn in totaal dus 18 meetpunten.
De nummers 1, 2, 3 corresponderen met de verschillende waarden per factor.
Er is hier sprake van een 'full factorial design'.

4.6 VALIDATIE VAN HET GEKOZEN MODEL M.B.T. DE PLAUSIBILITEIT

De validatie van het model is voornamelijk van belang m.b.t. de gekozen opnamepolitiek.

1. **Het beleid in het weekend.** Om organisatorische redenen, zoals het personeelsaspect, is het aannemelijk dat men de werkzaamheden in het weekend zoveel mogelijk zal willen beperken. Daarom zal men dus in het weekend zo weinig mogelijk patiënten opnemen. (Overigens wordt tegenwoordig in steeds meer ziekenhuizen ook op zondagmiddag opgenomen, omdat door de rust die er dan meestal is de patiënt dan beter op zijn behandeling voorbereid kan worden.) Op zaterdag wordt wel ontslagen, omdat de patiënt anders een heel weekend langer in het ziekenhuis moet blijven. Mensen die op zondag ontslagen moeten worden, moeten echter tot maandag in het ziekenhuis blijven. Het ontslaan op zaterdagochtend heeft minder consequenties dan ontslaan op zondag, omdat de personeelsbezetting op zaterdag waarschijnlijk hoger zal zijn dan op zondag.

2. **Afspraken.** Voor de voor- en nadelen van het wel of niet werken met afspraken wordt verwezen naar Hoofdstuk 1.

3. **Spoedbedden.** Er is gekozen voor het vrijhouden van enige bedden voor spoedpatiënten, omdat anders de kans te groot wordt dat een spoedpatiënt moet worden doorgestuurd. En het doorsturen van een spoedpatiënt is een vervelende gebeurtenis, met soms zelfs noodlottige gevolgen.

4. **Beddengebruik.** Elke afdeling gebruikt uitsluitend zijn eigen bedden. Dit heeft als nadeel dat er meer spoedpatiënten moeten worden doorgestuurd, omdat als een afdeling volledig bezet is de patiënt altijd wordt doorgestuurd, ook al is er elders een bed vrij.
Wel uitwisselen van de bedden heeft echter ook nadelen. Ten eerste vereist het extra organisatorische maatregelen. Verder kunnen er als een bed steeds naar de juiste afdeling wordt gebracht ruimteproblemen ontstaan. Als daarentegen de patiënt altijd behandeld wordt op de afdeling waar het bed eigenlijk thuis hoort, dan is het goed denkbaar dat bij deze behandeling moeilijkheden kunnen optreden.

5. **Opnamepolitiek 1.** Deze politiek, met vaste afspraaktermijnen per afdeling, is toepasbaar als er een eenvoudige afsprakenpolitiek nodig is en het ziektebeeld van de patiënten op een bepaalde afdeling niet sterk varieert.

6. **Opnamepolitiek 2.** Deze politiek, waarbij gewerkt wordt met prioriteiten, sluit waarschijnlijk beter bij de praktijk aan dan politiek 1. Kunnen sommige patiënten slechts beperkte tijd op de wachtlijst staan, anderen kunnen geruime tijd wachten voordat zij worden opgenomen.

7. **Opnamepolitiek** 3. Deze politiek, waarbij de afspraaktermijn afhankelijk is van de wachtrij en de bezetting op het moment van de afspraak, is vrij complex. Bij elke afspraak moet immers gekeken worden hoe lang de wachtrij is en moet de bezettingsgraad worden bepaald. Toch zijn er goede redenen voor aan te wijzen. Als op het moment de bezettingsgraad aan de hoge kant is, dan is de kans groot dat de mensen die nu op de wachtlijst staan langer moeten wachten en dan is het beter om voor de mensen die nu een afspraak maken vast een langere afspraaktermijn dan normaal aan te houden. Dezelfde redenering geldt voor een wachtrij die langer is dan normaal het geval is.

Hoofdstuk 5 TACTISCHE PLANNING

In dit hoofdstuk worden de invoergegevens en de waarden van de factoren gekwantificeerd. Ook worden de statistisch relevante zaken die met de experimenten verband houden, zoals de runlengte, steekproefomvang, startwaarden e.d. bepaald.

5.1 DE INVOERGEGEVENS

Om het geconstrueerde model te kunnen gebruiken zijn er allerlei invoergegevens nodig, die op de een of andere manier de 'real world data' weergeven. Deze gegevens kunnen op drie verschillende manieren worden ingevoerd.

1. Een theoretische verdeling. Het model gebruikt de gegevens indirect, d.w.z. door middel van een trekking uit een bepaalde verdeling.
2. Een empirische verdeling. D.w.z. het model gebruikt verdelingen die in het model zelf gespecificeerd worden.
3. Direct gebruik door het programma. Deze methode houdt in dat aan bepaalde programmavariabelen constante waarden worden toegevoegd.

In hoofdstuk 4 zijn zeven factoren onderscheiden. Deze moeten aan een invoeranalyse worden onderworpen. Volgens bovenstaande indeling ontstaat dan het volgende beeld:

1. Theoretische verdeling
   - patiëntenaanbod
   - ligduur van de patiënten
2. Empirische verdeling
   - gedeelte van de week
   - verdeling patiënten over de verschillende afdelingen
   - verdeling patiënten in spoed/niet-spoed
3. Direct gebruik
   - aantal bedden
   - opnamepolitiek.

5.2 HET PATIENTENAANBOD

Om het juiste aantal patiënten te genereren moet een bepaling gemaakt worden van het aankomstpatroon van deze patiënten over de week, over de dag en over de afdelingen.

Men verwacht per jaar in totaal 4.000 (40% van 10.000) klinische patiënten. Hiervan zal gemiddeld 40% als spoedpatiënt moeten worden opgenomen. Een uitzondering vormt de afdeling Chirurgie II, deze afdeling kent alleen spoedpatiënten. Het patiëntenaanbod voor deze afdeling bedraagt 13% van het totale aanbod van klinische patiënten. Op basis van deze gegevens kunnen de volgende berekeningen worden gemaakt.

- (87% × 4.000 × 60%)/(52 × 5) = 8,03 niet-spoedpatiënten per werkdag
- (13% × 4.000 + (87% × 4.000 × 40%))/365 = 5,24 spoedpatiënten per dag.

Op een werkdag komen dus is totaal 13,27 patiënten aan, op een weekenddag 5,24.

Vervolgens moet de verdeling van deze patiënten over de dag bepaald worden. Hiervoor hebben wij aangenomen dat de 5,24 spoedpatiënten over het hele etmaal gelijk verdeeld aankomen. Verder is aangenomen, dat de 8,03 gewone patiënten aankomen in de periode tussen 8.00 uur 's ochtends en 17.00 uur 's middags (d.w.z. in die periode van 9 uur kunnen ze terecht om een afspraak te maken).

Daarom wordt elke werkdag verdeeld in een dagperiode (8.00-17.00, de dag begint om 8.00) en een nachtperiode (17.00-8.00).

5.3 DE BEDDEN

Op soortgelijke wijze worden berekeningen uitgevoerd omtrent het aantal bedden, dat men als factor wil gebruiken. Uit 4.4 volgt het getal 160; een nadere berekening levert echter het volgende beeld:

afdeling	beddenaantal	extra bedden voor spoedgevallen
intern	53	1
chirurgie I	38	1
chirurgie II	29	-
gynaecologie	41	2
pediatrie	16	1
K.N.O.	6	1
	183	6

Voor deze factor worden nu de waarden 190 en 210 gekozen.

5.4 DE OPNAMEPOLITIEK

Ten aanzien van de afspraaktermijnen is na analyse van de gegevens gekozen voor de volgende termijnen:

afdeling	Afspraaktermijn
intern	14 dagen
chirurgie I	14 dagen
chirurgie II	niet van toepassing
gynaecologie	7 dagen
pediatrie	14 dagen
K.N.O.	7 dagen

5.5 DE STARTCONDITIES

Zoals in de systeemdefinitie bij de uitgangspunten is aangegeven, is het de bedoeling het ziekenhuisproces te simuleren onder normale omstandigheden, d.w.z. onder typische 'day-to-day operating conditions'. Bij de opzet van het simulatieprogramma zal dan ook rekening moeten worden gehouden met startcondities. Daarvoor bestaan in het algemeen drie mogelijkheden:

1. een zodanige runtijd kiezen, dat relatief gezien de nadelen van het beginnen met een leeg ziekenhuis worden opgeheven;
2. het begin van de run buiten beschouwing laten;
3. het kiezen van startcondities die (min of meer) typerend zijn voor een 'steady-state' situatie.

Bij het ziekenhuis is gekozen voor een combinatie, typerende startcondities èn een startperiode die buiten beschouwing wordt gelaten.

Om een 'steady-state' toestand te bereiken, zal tijdens de aanloopperiode het gehele proces van afspraken maken, opnemen en ontslaan moeten worden doorlopen. Tijdens de aanloopperiode wordt een bepaalde toestand gecreëerd, die dan als uitgangspunt dient voor de uiteindelijke simulatieperiode.

De problemen bij het bepalen van typerende startcondities zijn: het bepalen van de bezettingsgraad, van de ontslagdag van de patiënten die bij de start in het ziekenhuis liggen en het vaststellen welke afspraken er reeds voor de toekomstige periode zijn gemaakt.

1. De bezettingsgraad

De bezettingsgraad die als startsituatie van de simulatie kan worden genomen, kan op dezelfde manier worden bepaald als de verwachte bezettingsgraad bij opnamepolitiek 3.

2. Bepalen van de ontslagdagen

Dit is een moeilijke opgave. In feite kunnen al op de eerste simulatiedag mensen worden ontslagen. Als we nu voor de ligduur van de patiënten een trekking nemen uit de negatief exponentiële verdeling dan zullen op de eerste

dagen, voor de meeste afdelingen slechts weinig mensen worden ontslagen. Bovendien ontstaat er zodra de gemiddelde ligduur verstreken is een ontslaggolf, die niet volgens het normale patroon is, zodat in de eerste weken het beeld erg verstoord is. Om dit te ondervangen gebruiken we voor de ligduur een trekking uit een uniforme verdeling. Deze verdeling heeft als grenzen: 0 dagen en een bepaalde bovengrens. Het bepalen van deze bovengrens heeft erg veel problemen opgeleverd. Verschillende berekeningen zijn gemaakt, maar geen van allen leverden ze bruikbare mogelijkheden op. Zo werd bijvoorbeeld geprobeerd de grens zodanig te bepalen, dat van 95% van de patiënten die bij de start in het ziekenhuis lagen, de ontslagdag normaal gezien voor die grens zou liggen.

Tenslotte is gekozen voor een methode die ook niet helemaal juist is, maar een eind in de goede richting gaat. Deze methode neemt als uitgangspunt aan dat er elke dag gemiddeld evenveel patiënten worden ontslagen; deze veronderstelling lijkt niet onaanvaardbaar. Als nu bepaald kan worden hóeveel mensen er per dag ontslagen worden, dan kan door dat aantal te delen op het aantal mensen dat bij de start in het ziekenhuis ligt, het aantal 'ontslagdagen' bepaald worden.

De redenering luidt als volgt:

- aantal mensen dat op de afdeling ligt = ρ × aantal bedden

$$= \frac{\lambda}{n \cdot \mu} \times n = \frac{\lambda}{\mu}$$

- gemiddeld worden per dag $\lambda/365$ mensen opgenomen, per week dus 7 maal zoveel, die worden in 6 dagen ontslagen, dus per dag $\lambda/(365 \times {}^6/_7)$
- de bovengrens wordt dan

$$\frac{\lambda}{\mu} / \frac{\lambda}{(365 \times {}^6/_7)} = \frac{365}{\mu} \times {}^6/_7 \;.$$

Voor de 6 afdelingen is het resultaat van de berekening:

afdeling	bovengrens
intern	31,4
chirurgie I	19,1
chirurgie II	19,1
gynaecologie	11,3
pediatrie	23,6
K.N.O.	2,6

De berekening klopt niet helemaal, omdat er in de dagen volgende op de start ook nog mensen kunnen worden opgenomen die op dagen vóór de bovengrens worden ontslagen. Bij gebrek aan beter is echter toch maar met deze methode gewerkt.

3. Reeds gemaakte afspraken

Er moet bepaald worden hoeveel afspraken er al gemaakt zijn op $t = 0$ en voor wanneer. De periode waarvoor reeds afspraken zijn gemaakt, is niet zo moeilijk te bepalen, dat is namelijk steeds de afspraaktermijn. Doordat voor de afspraaktermijnen steeds gewerkt wordt met veelvouden van 7 kan

voor de bepaling van het aantal afspraken per dag gekeken worden naar het aantal niet-spoedpatiënten, dat elke dag een afspraak komt maken voor de betreffende afdeling.

Onder andere vanwege tijdgebrek, maar ook omdat het effect van eventuele verschuivingen e.d. moeilijk te voorspellen is, is voor iedere opnamepolitiek met dezelfde starttoestand gewerkt. Om dit effect enigszins weg te werken en vanwege de opmerking onder punt 2 is, ondanks de gecreëerde starttoestand toch nog gekozen voor een niet beschouwde beginperiode van drie weken. Omdat van geen enkele afdeling de afspraaktermijn langer is dan 14 dagen zijn dan in elk geval alle gemaakte afspraken aan de beurt geweest.

5.6 SIMULATIEPERIODE

In paragraaf 4.4 is reeds de in de opgave opgenomen simulatieperiode van 10 weken becommentarieerd. We gebruiken in deze simulatie één voortgezette run. Het aantal aankomende en behandelde patiënten ligt 'vanzelf' vast door deze keuze van de periode. Een validatie-aktiviteit zou dan moeten zijn, dat het gevonden aantal wordt getoetst aan de overwegingen van een steekproefgrootte afgezet tegen een gewenste betrouwbaarheid.

De waarnemingsperiode is een dag.

5.7 UITVOERANALYSE

De structuur van de uitvoer zal als volgt moeten zijn:

1. Kop

   Hierin wordt aangegeven wat het probleem is en wat de doelstelling van het simulatiemodel. Tevens wordt vermeld wat de startperiode en de uiteindelijke simulatieperiode is.

2. Per variant

   - beschrijving van de variant.
   - bezettingsgraad en wachtrijlengte per afdeling, per dag; deze gegevens worden per afdeling in een histogram naast elkaar weergegeven.
   - verdeling van de patiënten over de afdeling; voor elke afdeling worden de aantallen van een aantal onderscheiden groepen patiënten gegeven.
   - de gemiddelde wachtrijlengte en de gemiddelde bezettingsgraden van de afdelingen, de laatste onderscheiden naar werkdagen, zaterdagen en zondagen; met hun standaardafwijkingen.
   - gemiddelde ligduur en gemiddeld aantal wachtdagen van de patiënten per afdeling; met hun standaardafwijkingen.

5.8 HET RESULTATENONDERZOEK

Om te voldoen aan de vraag van de opdrachtgever zal een uitvoerig statistisch onderzoek gedaan moeten worden om elke opnamepolitiek te kunnen vergelijken, gegeven de overige condities. In deze case wordt dit onderzoek niet verder uitgewerkt.

Hoofdstuk 6 PROGRAMMASTRUCTUUR EN VALIDATIE

6.1 PROGRAMMAOPZET*)

De taak van het programma is om uitgaande van

1. het aantal bedden,
2. de beddenverdeling,
3. een opnamebeleid,

het ziekenhuissysteem een aantal dagen te simuleren en gegevens hierover te presenteren en dit voor een aantal varianten te herhalen.

De totaal te simuleren periode valt uiteen in 3 gedeelten:

1. **Vullen.** In dit gedeelte wordt het ziekenhuis gevuld met patiënten, zodanig dat elke afdeling zijn verwachte bezettingsgraad heeft. Van elke patiënt wordt de ontslagdag bepaald op de wijze zoals dat in hoofdstuk 5 is beschreven.
2. **Startperiode.** In deze periode worden de mensen die eerder een afspraak maakten opgenomen, patiënten worden ontslagen en afspraken worden vastgelegd. Aan het einde van de startperiode worden verschillende variabelen, zoals totaal aantal: spoedpatiënten, niet-spoedpatiënten, doorgestuurd aantal, op nul gezet.
3. **Uiteindelijke simulatieperiode.** Het vervolg van de startperiode.

Op één dag onderscheiden we de volgende belangrijke gebeurtenissen:

1. Patiënten komen aan:
   - poliklinisch = behandelen
   - klinisch = - spoed: zo mogelijk opnemen
- niet-spoed: afspraak maken
2. Patiënten worden opgenomen:
   - een bed wordt bezet
   - de ontslagdag wordt bepaald
3. Patiënten worden ontslagen:
   - er komt een bed vrij.

---

*) Dit hoofdstuk is sterk ingekort: de detailspecificaties en het computerprogramma zijn niet opgenomen.

Tijdens deze dag moeten we een aantal gegevens bijwerken die nodig zijn om de gewenste uitvoer te krijgen, zoals deze bij de outputanalyse is aangegeven.
We hebben in ons programma te maken met 3 belangrijke componenten:

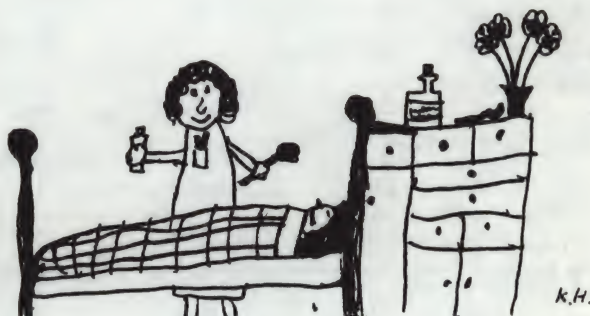
- ziekenhuis
- afdeling
- patiënt

Omdat geprogrammeerd wordt in de simulatietaal Simula, kunnen we deze componenten aangeven d.m.v. classes. Naast een PSD van het hoofdprogramma worden in de volgende paragrafen de PSD's gegeven van de:

- class AFDELING
- class PATIENT
- class ZIEKENHUIS.

6.2 PSD HOOFDPROGRAMMA

```
declaraties
inlezen
initialisatie
KOP
for bedaant:=190,210 do
    for opnpol:=1,2,3 do
        u-waarden voor elke variant inlezen
        hospital:-new ZIEKENHUIS (bedaant,opnpol)
        RESULTAAT (simper)
```



6.3 PSD CLASS ZIEKENHUIS

```
declaratie
INIT EN STARTCOND (bedaant)
hold (8) {de dag begint om 8 uur}
for week:=1 step 1 until simper + startper do
    if week = startper + 1
    T: WEGVEGEN
    F: 
    for dag:=1 step 1 until 7 do   {za=6, zo=7}
        dezedag:=dezedag + 1
        if dag ≠ 7
        T: ONTSLAAN (dezedag)
        F: 
        if dag ≠ 6 ∧ dag ≠ 7
        T: spoedper:=false
           eindtijd dag bepalen
           tak:=negexp (1/takdag,u)
           hold (tak)
        F: 
        while time < eindtijd do
            VOLGENDE PATIENT (takdag,eindtijd)
        spoedper:=true
        eindtijd nacht bepalen
        tak:=negexp (1/takspoedper,u)
        hold (tak)
        while time < eindtijd do
            VOLGENDE PATIENT (takspoedper,eindtijd)
        BEPAALBEZGRAAD (dag)
```

6.4 VALIDATIE VAN HET PROGRAMMA

TESTEN

Het testen van een programma bestaat uit 3 delen:

1. Het toetsen op syntactische fouten. Dit wordt gedaan door de compiler.
2. De toets op het niet toelaatbaar zijn van bepaalde operaties.
3. De controle van het programma op logische fouten.

Ad 2.

Deze toets, die ook door de compiler wordt uitgevoerd, resulteerde in een tweetal foutmeldingen, die beide betrekking hadden op de berekening van gemiddelden en standaardafwijkingen in RESULTAAT. De eerste maal betrof het een deling door nul als gevolg van een foutieve berekening van een standaardafwijking. Deze foutmelding was het gevolg van het feit dat wij een standaardafwijking van de totale bezetting van het gehele ziekenhuis wilden berekenen, maar deze bleek negatief te zijn. Deze berekening is daarom uit het programma weggelaten.

Ad 3.

Op een tweetal manieren hebben we ons programma gecontroleerd op logische fouten. Allereerst door het laten afdrukken van een aantal elementaire variabelen op een aantal relevante punten. Vervolgens door het controleren van enkele berekeningen met behulp van deze tussenuitvoer. Afgedrukt werden o.a.:

- aantal ontslagen patiënten
- aantal opgenomen gewone patiënten
- aantal opgenomen spoedpatiënten
- de wachtrij
- ligduur van de patiënten
- de bezettingsgraad,

meestal werden deze gegevens per dag afgedrukt.

Met deze tussenuitvoer werden de berekeningen van de gemiddelden en standaardafwijkingen gecontroleerd.

Bij het testen bleek o.a. dat de wachtrij veel te hoog was. Dit werd veroorzaakt door een fout in de periode-indeling. In plaats van de nachtperiode te laten lopen van 17.00 tot de volgende morgen 8.00 liep de nachtperiode maar tot 24.00 uur.

Daardoor kwamen er ook 's nachts patiënten aan om een afspraak te maken. Het aantal afspraakpatiënten werd hierdoor veel te groot en de wachtlijst werd steeds langer.

Een andere vervelende fout die het testen opleverde, was dat er mensen bleken te zijn waarvan de ligduur nul was. In beginsel is dit natuurlijk mogelijk, de patiënten kunnen bijvoorbeeld direct overlijden bij aankomst in het ziekenhuis. In ons programma levert dit echter problemen op, omdat we de patiënten aan het begin van de dag ontslaan. Zo wordt een patiënt die op zijn aankomstdag het ziekenhuis al weer moet 'verlaten', nooit meer ontslagen.

Daarom kreeg de ligduur de waarde 1 als een trekking de waarde nul opleverde.

In eerste instantie begon het programma steeds met een leeg ziekenhuis. Toen in een later stadium als uitgangspunt een startsituatie werd gecreëerd bleek dat de wachtrij negatief was. De oorzaak werd gevonden in de afspraken die bij het starten genegeerd werden. Deze mensen werden later natuurlijk opgenomen en de wachtrij werd automatisch verminderd. Het bleek echter dat het ophogen van de wachtrij bij de startsituatie achterwege was gebleven en dat hier dus de oorzaak van de fout lag.

Hoofdstuk 7 VALIDATIE EN RESULTATENANALYSE

7.1 VERIFICATIE

Verificatie wil zeggen dat men onderzoekt of het simulatiemodel zich gedraagt zoals de onderzoeker dat bedoeld heeft. In hoofdstuk 6 'Testen' is reeds een groot gedeelte van deze verificatie beschreven. Door het laten afdrukken van detailgegevens per dag of per patiënt kon bekeken worden of alles in het programma correct verliep. Na het verwijderen van een aantal fouten bleef er toch nog steeds iets over dat niet klopte. Dat was afdeling 6, K.N.O.; in deze afdeling werden veel te weinig patiënten behandeld en de wachtrij werd steeds langer.

De oorzaak hiervan was moeilijk te vinden. Tenslotte bleek dat de fout lag in de procedure ONTSLAAN. Bepaalde patiënten werden overgeslagen bij het nagaan van de ontslagdag en kwamen er daardoor nooit meer uit.

7.2 VALIDATIE VAN EEN AANTAL UITVOERGEGEVENS

Over een aantal gegevens die het programma als doel van de simulatie moest opleveren, is een uitspraak te doen t.a.v. wat men van deze gegevens verwacht. Dit wordt bekeken voor de variant met 190 bedden en opnamepolitiek 1.

1. De gemiddelde ligduur

De gemiddelde ligduur is een typische variabele in het model. Enerzijds wordt dit gemiddelde voor elke afdeling als invoervariabele op een bepaalde waarde gesteld. Anderzijds is de gemiddelde ligduur als uitvoervariabele gewenst. Men zou dus verwachten dat deze in principe aan elkaar gelijk zijn. Dit bleek echter niet zo te zijn. Als eerste oorzaak hiervoor kan de verstorende werking die van het weekend uitgaat worden aangewezen. Patiënten die op zondag ontslagen zouden moeten worden mogen pas maandag het ziekenhuis verlaten, zij krijgen daardoor dus een hogere ligduur. Verder is telkens als de trekking voor een ligduur 0 was de waarde 1 ingevuld; ook dit kan, vooral voor afdelingen met een lage ligduur, verstorend werken.

Toch bleek de gemiddelde ligduur voor de meeste afdelingen lager dan het ingevoerde gemiddelde te zijn. De oorzaak hiervan moet gezocht worden in de manier waarop het gemiddelde wordt bepaald. Het is namelijk de

gemiddelde ligduur van de patiënten die tijdens de startperiode of later zijn opgenomen èn in de simulatieperiode zijn ontslagen. Patiënten met een lange ligduur die niet in deze periode ontslagen worden tellen dus niet mee. Daardoor valt het gemiddelde lager uit, dit heeft vooral invloed op de afdelingen met een lange ligduur, zoals afdeling 1, Intern.

2. **Het gemiddeld aantal wachtdagen**

Voor elke afdeling geldt een bepaalde afspraaktermijn. Deze kan dienen als vergelijkingsmateriaal t.a.v. het gemiddelde aantal wachtdagen per patiënt. Gemiddeld zal het aantal wachtdagen van een patiënt gelijk zijn aan deze afspraaktermijn. Verschillen kunnen optreden als de patiënt niet op de dag dat hij heeft afgesproken, kan worden opgenomen en dus langer moet wachten. Doordat als afspraaktermijn steeds een veelvoud van 7 is gekozen gaat er geen verstorende werking van het weekend uit. (Overigens zal bij de andere politieken de vergelijking moeilijker zijn, omdat daarbij de afspraaktermijn niet voor elke patiënt gelijk is, maar afhankelijk is van zijn prioriteit of van de 'toestand' van de afdeling.) De volgende vergelijking kan worden opgesteld:

Afdeling	Afspraaktermijn	Gemiddelde wachttijd
Intern	14 dagen	14.00 dagen
Chirurgie I	14	14,01
Chirurgie II	--	--
Gynaecologie	7	12,35
Pediatrie	14	14,16
K.N.O.	7	15,14

Bij een vergelijking van de afspraaktermijnen met de gevonden gemiddelden blijkt dat alleen de afdeling Gynaecologie en de afdeling K.N.O. afwijkende gemiddelden opleveren. De verklaring hiervoor in het geval van de afdeling Gynaecologie is waarschijnlijk dat deze afdeling 2 bedden kent die moeten worden gereserveerd voor spoedpatiënten en dat als gevolg van dit grotere aantal de niet-spoedpatiënten eerder worden verschoven, wat leidt tot een hoger aantal wachtdagen.

De afwijking van de afdeling K.N.O. kan waarschijnlijk worden verklaard door het feit dat het éne spoedbed van deze afdeling een groot percentage van het totaal aantal bedden van deze afdeling (slechts 6) in beslag neemt. Dit heeft tot gevolg dat het aantal verschoven patiënten en daarmee het gemiddelde aantal wachtdagen toeneemt. (Met verschoven patiënten worden patiënten bedoeld die niet op hun afspraakdag kunnen worden opgenomen, omdat er geen bed beschikbaar is.)

3. **De gemiddelde wachtrijlengte**

De wachtrij die voor een bepaalde afdeling verwacht wordt, kan berekend worden op basis van het patiëntenaanbod en de afspraaktermijn voor de betreffende afdeling. Op dag x staan er mensen op de wachtlijst die voor een bepaalde dag in de toekomst hebben afgesproken, voor zover als de afspraaktermijn reikt. Het aantal mensen dat per dag een afspraak komt maken vermenigvuldigd met de afspraaktermijn (gecorrigeerd voor de weekenddagen) geeft dan de normale wachtrijlengte. De volgende vergelijking kan worden opgesteld:

Afdeling	Verwachte wachtrij	Gevonden wachtrij
Intern	15,9	12,87
Chirurgie I	18,0	14,83
Chirurgie II	--	--
Gynaecologie	16,4	24,91
Pediatrie	6,4	6,46
K.N.O.	9,5	16,47

Ter verduidelijking volgt hieronder de berekening voor de afdeling Intern, de andere berekeningen verlopen op dezelfde manier.

$$((\frac{0,15}{1 - 0,13} \times 4.000 \times 0,6) \times (14 - 2 \times 2) = 15,9$$

De afwijking tussen de verwachte wachtrij en de gevonden wachtrij is voor de afdeling Gynaecologie en de afdeling K.N.O. waarschijnlijk op dezelfde manier te verklaren als de afwijking die er bij de wachtdagen optrad. Het aantal spoedbedden is relatief hoog en dat resulteert in een groter aantal verschoven patiënten. Deze verschoven patiënten blijven dus langer op de wachtlijst staan dan de bedoeling was en daardoor wordt de wachtrij langer.

De afwijking naar beneden van de afdelingen Intern en Chirurgie I is moeilijker te verklaren. Een afwijking naar boven is te verklaren door de verschuivingen die er plaatsvinden. De enige mogelijkheid is dat er minder patiënten zijn aangekomen voor de afdeling of dat het percentage spoedpatiënten boven de 40% ligt, omdat er sprake is van een stochastisch model. Bij berekening blijken deze percentages inderdaad iets naar beneden af te wijken.

7.3 RESULTATENONDERZOEK

7.3.1 Betrouwbaarheid van de schattingen

Bij het bepalen van het functioneel model is al aangegeven dat de bepaling van het 'datum point' problemen opleverde, omdat de kans groot is dat er tussen de steekproefuitkomsten autocorrelatie bestaat.

We hebben de keuze tussen een steekproefgrootte (n) van 1 (70 dagen), 10 (weken), 70 (dagen). Bij deze laatste twee is er sprake van autocorrelatie als gevolg van het karakter van het ziekenhuisgebeuren.

De keuze van een steekproefgrootte van 1 heeft echter wel tot gevolg dat de betrouwbaarheidsintervallen die we van de resultaten kunnen opstellen aanzienlijk groter zijn en het is dan ook de vraag of een dergelijk groot interval nog zin heeft. Het opstellen van een betrouwbaarheidsinterval is bovendien moeilijk, omdat de populatievariantie niet bekend is. Om dan toch een betrouwbaarheidsinterval te kunnen opstellen, is een grotere n dan 50 nodig, i.v.m. de benadering door de normale verdeling. We zullen daarom alleen een betrouwbaarheidsinterval opstellen voor een n van 70. We moeten er dan wel rekening mee houden dat door de autocorrelatie de geschatte variantie kleiner is dan de werkelijke variantie. Voor een zuiver betrouwbaarheidsinterval is een aantal herhalingen van de '10 weken' nodig.

Een 95%-betrouwbaarheidsinterval ziet er als volgt uit:

$$\bar{x} - \frac{\sigma}{\sqrt{70}} \cdot 1,96 \leqq \mu \leqq \bar{x} + \frac{\sigma}{\sqrt{70}} \cdot 1,96$$

Voor de over de week gemiddelde bezettingsgraden van de zes afdelingen zien de betrouwbaarheidsintervallen er als volgt uit (voor de gegevens van de eerste variant):

Afdeling	Betrouwbaarheidsinterval
Intern	0,71 ± 0,0164
Chirurgie I	0,86 ± 0,0211
Chirurgie II	0,77 ± 0,0398
Gynaecologie	0,91 ± 0,0258
Pediatrie	0,69 ± 0,0375
K.N.O.	0,90 ± 0,0328

7.3.2 Resultatenanalyse

In deze paragraaf wordt aan het uiteindelijke doel van het model toegekomen, d.w.z. de alternatieven worden met elkaar vergeleken.
De gegevens die voor vergelijking in aanmerking komen zijn:

- aantal doorgestuurde patiënten
- bezettingsgraden, gemiddeld over de week
- het gemiddelde aantal wachtdagen

Deze gegevens zullen daartoe eerst in een aantal staatjes worden weergegeven waarna een algehele vergelijking kan worden opgezet.

Aantal doorgestuurde patiënten

	190 bedden			210 bedden		
Afdeling	opn-pol 1	opn-pol 2	opn-pol 3	opn-pol 1	opn-pol 2	opn-pol 3
1	0	0	0	0	0	0
2	0	2	2	0	0	0
3	21	21	10	4	14	1
4	5	9	6	4	2	3
5	1	0	0	0	1	0
6	22	23	19	9	16	18
Totaal	49	55	37	19	33	22

Weekgemiddelde van de bezettingsgraden

	190 bedden			210 bedden		
Afdeling	opn-pol 1	opn-pol 2	opn-pol 3	opn-pol 1	opn-pol 2	opn-pol 3
1	0,76	0,75	0,79	0,76	0,70	0,71
2	0,87	0,91	0,88	0,82	0,74	0,87
3	0,79	0,84	0,80	0,79	0,77	0,68
4	0,88	0,87	0,90	0,81	0,82	0,80
5	0,52	0,59	0,65	0,70	0,77	0,73
6	0,93	0,92	0,88	0,82	0,85	0,84

Het gemiddelde aantal wachtdagen

	190 bedden			210 bedden		
Afdeling	opn-pol 1	opn-pol 2	opn-pol 3	opn-pol 1	opn-pol 2	opn-pol 3
1	14,00	14,24	14,10	14,00	14,24	14,19
2	14,00	14,55	14,28	14,00	14,24	15,22
3	--	--	--	--	--	--
4	7,60	7,07	11,61	7,23	6,81	11,00
5	14,04	12,25	14,04	14,00	12,82	14,07
6	20,72	11,73	13,03	8,21	12,20	14,64

Voor een analyse van deze resultaten zal men primair kijken naar de belangrijkste criteria. Het aantal doorgestuurde patiënten speelt in dit verband een hoofdrol.

Doorgestuurde patiënten

Bij 190 bedden valt op dat het aantal doorgestuurde patiënten bij politiek 3 het laagste is. Opnamepolitiek 3 houdt in dat bij afspraken steeds gekeken wordt naar de bezettingsgraad en de al aanwezige wachtrij. Blijkbaar wordt de afspraaktermijn dus gemiddeld langer, de afspraakpatiënten moeten langer wachten, maar er kunnen meer spoedpatiënten worden opgenomen. Als men naar het gemiddelde aantal wachtdagen kijkt blijkt dit voor de meeste afdelingen bij politiek 3 inderdaad iets hoger te zijn, maar niet overtuigend.

Een vergelijking van het aantal van 190 met het aantal van 210 bedden laat zien dat voor alle drie opnamepolitieken het aantal doorgestuurde spoedpatiënten afneemt. Bij 210 bedden geven zowel politiek 1 als 3 een gering aantal doorgestuurde patiënten te zien.

Vergelijking van politiek 1 met politiek 2

Deze strategieën zijn in principe eigenlijk gelijk. Ze hebben gemiddeld dezelfde afspraaktermijn. Alleen de afspraakpatiënten zullen een verschil merken. Toch blijkt dat het aantal doorgestuurde patiënten bij politiek 2 bij 210 bedden hoger ligt dan bij politiek 1. Dit kan komen, doordat er een aantal van de afspraakpatiënten eerder wordt opgenomen, namelijk die met prioriteit 1, dan bij politiek 1.

T.a.v. de bezettingsgraad kan worden opgemerkt dat voor de meeste afdelingen het verschil tussen politiek 1 en politiek 2 niet erg groot is.

Vergelijking van politiek 1 met politiek 3

Omdat politiek 1 en 2 in wezen niet veel verschillen zal worden volstaan met een vergelijking van politiek 1 en 3. Eerder is al opgemerkt dat het aantal doorgestuurde patiënten bij politiek 3 in totaal lager is. Bij een nadere analyse blijkt ook hier weer dat een vergelijking voor de afzonderlijke afdelingen verschillend uitvalt.

De verschillen in bezettingsgraad bij beide strategieën zijn voor de afzonderlijke afdelingen ook weer niet altijd gelijk. De laagste waarde die bij deze politieken gemeten is, is een bezettingsgraad van 68%, bij een aantal van 210 bedden. De vraag of dit percentage niet te laag is, is zonder verdere gegevens over de criteria die men hiervoor in de ziekenhuiswereld hanteert moeilijk te beantwoorden.

7.4 CONCLUSIES

Uit de resultatenanalyse in de vorige paragraaf kan worden geconcludeerd dat er geen duidelijke aanbeveling kan worden gegeven voor het beleid dat de beste resultaten oplevert. Dit is o.a. afhankelijk van de criteria die men t.a.v. de gegevens hanteert. Is het van het hoogste belang dat het aantal doorgestuurde patiënten zo laag mogelijk wordt gehouden, dan is het duidelijk dat gekozen moet worden voor opnamepolitiek 1 met een aantal van 210 bedden. Is de bezettingsgraad het belangrijkste criterium dan gaat de aanbeveling in de richting van opnamepolitiek 3, dan echter met 190 bedden.

Zoals al eerder is opgemerkt bestaan er voor de verschillende gegevens tussen de afdelingen onderling moeilijk te verklaren verschillen. Om tot een beter gefundeerd oordeel te komen is het daarom noodzakelijk de analyse per afdeling afzonderlijk te laten verlopen. Ook zou een aantal andere gegevens nader bekeken moeten worden.

Bedoeld wordt o.a. het aantal verschoven patiënten. Bij opnamepolitiek 2 zou onderscheid gemaakt moeten worden tussen de verschillende prioriteiten.

Verder zou steeds bekeken moeten worden hoe het aantal opgeschoven patiënten is opgebouwd, d.w.z. hoeveel patiënten worden 1 dag, hoeveel 2 dagen, enzovoort opgeschoven.

Ook lijkt het dat de werkwijze die is toegepast voor de verdeling van de bedden over de afdelingen misschien niet optimaal is. Vooral afdeling 6 valt hierbij op. Doordat deze afdeling een lage gemiddelde ligduur heeft, is het aantal bedden klein. Waarschijnlijk te klein, want bij alle varianten is het aantal doorgestuurde patiënten op afdeling 6 het hoogste, terwijl deze afdeling bijvoorbeeld toch minder patiënten hoeft op te nemen dan afdeling 4.

Er zou dus geëxperimenteerd moeten worden met een andere verdeling van de bedden over de afdelingen. Daarbij is het om tot een goed oordeel te komen waarschijnlijk ook beter om een groter aantal waarden t.a.v. het totale aantal bedden te onderscheiden.

Tenslotte

Met betrekking tot een eventuele gevoeligheidsanalyse kan tenslotte worden opgemerkt dat dit niet voor alle factoren even eenvoudig is. Dit vanwege de vele berekeningen die gemaakt zijn i.v.m. het bepalen van de invoergegevens en de startcondities.

Het patiëntenaanbod is bijvoorbeeld voor de klinische patiënten nu gesteld op 4.000 patiënten per jaar: dat getal veranderen brengt een groot aantal nieuwe berekeningen met zich mee.

Wel eenvoudig te veranderen factoren zijn bijvoorbeeld: de verdeling van de patiënten in prioriteitengroepen en hun diverse afspraaktermijnen, de 'kritieke waarden' van de bezettingsgraad en de wachtrij bij opnamepolitiek 3, de afspraaktermijnen bij politiek 1 en de gemiddelde ligduren.

BIJLAGE 1: χ^2-toets op "De ligduur van 200 chirurgie patiënten"

Getest wordt:
H_0: de ligduur $\underline{x}$ is exponentieel verdeeld met

$$P(\underline{x} \leq x) = - e^{-x/20} \quad \text{en}$$

$$E\underline{x} = 20$$

Aantal ligdagen	f_i	e_i
1 t/m 13	103	98,17
14 - 26	43	48,67
27 - 39	31	25,41
40 - 52	9	13,26
53 - 65	6	6,92
66 - 182	8	7,54
	200	119,97

f_i: de werkelijke frequentie
e_i: de theoretische frequentie

Hieruit volgt:

$$\chi^2 = \Sigma_i \frac{(f_i - e_i)^2}{e_i} = 3,65$$

De tabel geeft met 6-1-1 = 4 vrijheidsgraden en $\alpha = 0,05$ als kritieke waarde (c) de waarde 9,49.

Dus, omdat $\chi^2 < c$: H_0 niet verwerpen.

BIJLAGE 2: χ^2-toets op "Het gynaecologie-aanbod over 200 werkdagen"

Getest wordt:
H_0 : het aanbod $\underline{x}$ is Poisson verdeeld met

$$P(\underline{x} = k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!}$$

$$\lambda = 4{,}86$$

k	f_i		e_i	
0	2	6	1,55	9,08
1	4		7,53	
2	17		18,31	
3	29		29,66	
4	32		36,03	
5	34		35,02	
6	30		28,37	
7	19		19,70	
8	21		11,97	
9	9	12	6,46	9,60
10	3		3,14	

f_i: de werkelijke frequentie
e_i: de theoretische frequentie

Hieruit volgt:

$$\chi^2 = \Sigma_i \frac{(f_i - e_i)^2}{e_i} = 9{,}16$$

De tabel geeft met 9-1-1 = 7 vrijheidsgraden en $\alpha = 0{,}05$ als kritieke waarde (c) de waarde 14,1.

Dus, omdat $\chi^2 < c$: H_0 niet verwerpen.

BIJLAGE 3: "Norm voor het aantal bedden"

Het uitgangspunt voor het beddenaantal, gebruikt in dit model, is gebaseerd op een gedeelte uit het boek "De Economie en het Ziekenhuis"*). De passage die hierop betrekking heeft, luidt als volgt:

"In dit kader wordt even aandacht geschonken aan de norm voor het aantal bedden. De ziekenhuisgrootte wordt uitgedrukt in het aantal bedden. Hoe meer bedden, hoe groter het ziekenhuis. Het aantal bedden geeft weliswaar niet de totale capaciteit aan van het ziekenhuis, maar voorlopig is er geen betere maatstaf.

Sinds jaren wordt voor de beoordeling van plannen de beddennorm gehanteerd. Voor de eenvoud beperk ik mij tot de algemene ziekenhuizen. Voor deze categorie is de norm 4 pro mille, d.w.z. 4 ziekenhuisbedden per 1.000 inwoners. In de ziekenhuiswereld is fel geprotesteerd tegen deze norm, maar de moeilijkheid daarbij is dat de protesten niet 'hard' gemaakt kunnen worden. Het is wetenschappelijk niet aantoonbaar dat 4 bedden per 1.000 inwoners te weinig is om een goede ziekenhuisvoorziening te bieden. Een vergelijking met het buitenland gaat niet op, omdat de gezondheidszorg daar anders is georganiseerd. Men kan de beddennorm niet beoordelen zonder de poliklinische voorzieningen in de beschouwingen te betrekken.

De beddennorm van 4 pro mille moet men als politieke norm zien: Het Nederlandse volk is bereid zoveel geld aan de gezondheidszorg te besteden dat er voor iedere 1.000 inwoners 4 ziekenhuisbedden beschikbaar zijn."

(In hetzelfde boek staat behalve de beddennorm ook nog een opnamecoëfficiënt vermeld. De opnamecoëfficiënt voor algemene ziekenhuizen is 10%, d.w.z. per jaar worden 10 op de 100 mensen in het ziekenhuis opgenomen.)

---

*) De economie en het ziekenhuis, Drs. A. Mulder (uitg. De Tijdstroom).

HOOFDSTUK X

Opgaven

OPGAVE I

Een bevriende winkelier bij u in de buurt is bekend met uw bedrijfskundige studie en weet ook dat u een deskundige bent op het gebied van simulatie. Hij vraagt u om een onderzoek in te stellen naar zijn bedrijfsvoering. Na een aantal gesprekken met de winkelier stelt u de volgende punten vast.

1. Het betreft een typische eenmanszaak.
2. De openingstijden zijn van 9.00 - 18.00 uur, non-stop.
3. De aankomst van klanten heeft een kans van p per minuut. (Bijvoorbeeld p = 1/5, d.w.z. gemiddeld komt elke vijf minuten een klant binnen.)
4. 90% van de klanten wordt bediend met een helptijd welke uniform verdeeld is tussen a en b minuten (bijvoorbeeld U[2,10]).
5. 10% van de klanten echter willen het winkelbezoek wel eens als een welkome afleiding op het dagelijkse gebeuren beschouwen en zij houden de winkelier een tijd bezig, welke Poisson verdeeld is met een gemiddelde van 15, doch maximaal 60 minuten duurt en minimaal 2.
6. De gemiddelde (netto) winst per klant bedraagt $ G,-- (bijvoorbeeld $ 2,50).

U heeft afgesproken met uw winkelier om de volgende alternatieven te onderzoeken.

1. Het huidige systeem, zoals boven beschreven is.
2. De winkelier neemt een winkelbediende in loondienst; kosten $ F,-- per jaar (bijvoorbeeld $ 50.000,--).
3. Hij past een strategie toe, waarbij de kletsklanten na 10 minuten worden verzocht te vertrekken. De helft van deze klanten verliest hij aan de concurrentie.
4. De klanten, die voor zich Q (bijvoorbeeld 5) klanten zien wachten, vertrekken weer zonder geholpen te worden.
5. Die klanten, welke een helptijd onder de 4 minuten hebben, worden het eerst geholpen met de discipline van 'Shortest time first'!

Vragen en opdrachten*)

1. Voer een handsimulatie uit voor de eerste 6 klanten. Gevraagd wordt om de resultaten in een tabel weer te geven als volgt:

klant-nr.	aankomst-tijd klant	helptijd klant	wachttijd klant	vertrek-tijd	lengte wachtrij	voortschrijdend gemiddelde	
						wachttijd	helptijd
1							
2							
3							
4							
5							
6							

Over welke uitvoergegevens zou u verder nog graag willen beschikken?

2. Is het systeem (alternatief 1) statisch of dynamisch; deterministisch of stochastisch; een 'steady-state' systeem? Is hier sprake van een 'fixed-time-increment' model of van een 'next-event-time' model? Licht de antwoorden toe.

3. Zou men de term 'Monte Carlo experiment' mogen toepassen voor dit simulatie-onderzoek? Waarom (niet)?

4. Het systeem wordt bij elke run 'leeg' gestart. Is dit acceptabel? Noem mogelijkheden om de startproblematiek in het algemeen op te vangen.

5. Maak een globale berekening om p en de helptijd U[a,b] met elkaar in overeenstemming te brengen. (Ga uit van de numerieke gegevens.) Maak op grond van een redelijk jaarinkomen voor de winkelier een schatting voor een gewenste $ G,-- per klant. Bekijk op grond van deze beschouwingen de mogelijkheden of wenselijkheden van alternatief 2.

6. Bereken het aantal 'runs'. Neem zelf waarden aan voor eventueel ontbrekende gegevens. Neem de betrouwbaarheid op een waarde van 95%. Welke criteria legt u aan?

*) De beantwoording van de vragen dient steeds gepaard te gaan met een deugdelijke argumentatie.

7. Hoe kan een simulatierun beëindigd worden?
   a. In dit concreet geval.
   b. Welke technieken kunnen in het algemeen worden gebruikt?

8. Ontwerp een experimenteel model voor het simuleren van alternatief 4.

OPGAVE II

Wij beschouwen een eenvoudig gestileerd voorraadsysteem. De vraag D en de levertijd LT (uitgedrukt in dagen) zijn stochastische variabelen, met bekende waarschijnlijkheidsverdelingen, respectievelijk gegeven door $f_1(D)$ en $f_2(LT)$. De voorraad wordt dagelijks verlaagd met de totale vraaghoeveelheid van die dag. Wanneer het niveau gelijk aan of onder het bestelpunt BP komt, wordt een beslissing gegenereerd voor een bestelhoeveelheid Q* (optimale bestelhoeveelheid).

Bij levering van deze hoeveelheid wordt de voorraad bijgewerkt. De totale kosten TC van het systeem bestaan uit drie kostengroepen: TC_1, TC_2 en TC_3, waarbij de C_i kostenparameters voorstellen (meestal per stuk en/of per tijdseenheid).

Men wenst dit systeem te simuleren om de invloed te onderzoeken van de verschillende systeemparameters op de totale voorraadkosten TC. De volgende vragen hebben in principe betrekking op het beschreven voorraadsysteem. Geef echter waar mogelijk c.q. nodig op sommige plaatsen een wat uitvoeriger en meer in het algemeen geldend antwoord.

Om de vragen te kunnen beantwoorden acht u wellicht verdere veronderstellingen m.b.t. het voorraadsysteem nodig. U dient deze dan zorgvuldig te definiëren en te beargumenteren.

Vragen

1. Uit welke drie onderdelen (TC_1, TC_2, TC_3) bestaan de totale kosten van een voorraadsysteem?
   Definieer de parameters C_1, C_2 en C_3 en vergeet niet de dimensie vast te stellen.

2. Geef een visualisering van het voorraadsysteem m.b.v. een schets. Laat zien wat het systeem ingaat en wat er uitkomt. Catalogiseer de relaties, verdelingsfuncties, parameters en de verschillende variabelen.

3. Zoudt u behalve de genoemde doelstelling nog andere doelstellingen van het simulatiemodel en het simuleren kunnen formuleren?

4. Noem alle factoren die u met behulp van bovenstaande gegevens in een experimenteel ontwerp zou kunnen invoeren. Hebt u voldoende gegevens om een concreet experimenteel ontwerp te construeren?
   Zo niet: vul uw gegevens aan.

5. Stel dat van elke factor twee waarden in beschouwing worden genomen.
   U simuleert per keer het voorraadsysteem gedurende 180 dagen.
   Elke run wordt 30 maal herhaald om de fout van het 'random effect' te reduceren.
   Stel een volledig experimenteel ontwerp op.
   Hoeveel meetpunten krijgt u?

6. Wat betekent het getal: '180 dagen'. Hoe wordt dit genoemd?
   Geef een methode (korte omschrijving) om het getal '180' te bepalen.

7. Wat stelt de '30' van de herhalingen voor?
   Bespreek in verband hiermee het verschil tussen een 'replicated run' en een 'continued run' en geef van beide mogelijkheden aan waar de knelpunten zitten wat betreft:
   a. afhankelijkheid;
   b. voorloopperiode.
   Welke methode zou u voor het voorraadsysteem kiezen en waarom?
   Vindt u een sterke correlatie tussen de herhalingen wenselijk?
   Welke methode gaat u toepassen om deze correlatie te bewerkstelligen?

8. Zijn er redenen waarom u liever van het volledig experimenteel ontwerp zou afzien?
   Noem en bespreek ze kort.
   Welk alternatief kiest u dan? Werk dit uit.
   Geef de motivatie voor uw keuze en bespreek de optredende consequenties en nadelen.

9. Welke *startvoorwaarden* legt u aan het voorraadsysteem op?
   Motiveer! Geef aan hoe de *startwaarden* van de relevante parameters uit het voorraadsysteem bepaald worden.

10. Veronderstel dat u een geschikt experimenteel model hebt opgebouwd. Wat kunt u *globaal* gesproken doen om uw model te valideren? U gaat nu experimenteren met uw model en u verkrijgt na enige tijd een serie uitkomsten. Wat gaat u met deze uitkomsten doen?

11. Wat zullen voor de voorraadmanager de belangrijkste kenmerken zijn, voor acceptatie van uw systeem?
    Welke eisen zal hij stellen opdat uw systeem *ontwerp*-technisch en *programma*-technisch zal voldoen? (Apart behandelen.)

OPGAVE III

Een zogeheten thesaurie-informatiesysteem houdt zich o.a. bezig met het aantrekken van financieringsmiddelen, met cash-management, e.d. Het thesaurie-systeem kan als een stafsysteem t.b.v. het financieel informatiesysteem worden beschouwd.

Het management wordt ondersteund door dit thesaurie-informatiesysteem teneinde financieringsbeslissingen van het bedrijf goed gefundeerd te kunnen nemen. Ook zal m.b.v. dit systeem toezicht gehouden worden op bijvoorbeeld: de rentabiliteit van de investeringen, de kosten van eigen en vreemd vermogen, de uitstaande debiteuren, beleggingen, voorraden, enzovoort.

1. Leg uit in welke *betekenissen* simulatie het financiële management behulpzaam kan zijn. Denk hierbij aan de managementfuncties. Een voorbeeld uit het hierboven beschreven thesauriesysteem kan dit verduidelijken.

2. Om financiële beslissingen t.a.v. de toekomst te kunnen nemen, dient het simulatiemodel van de thesaurus 'on line' faciliteiten t.b.v. het management ter ondersteuning te bieden. Kunt u enige faciliteiten noemen?

3. Noem enige algemene eisen, die gebruikers aan een simulatiemodel kunnen en zullen stellen, dus ook aan het financiële simulatiemodel.

4. Om welke redenen zou hier zeker gevoeligheidsanalyse gebruikt moeten kunnen worden? Noem eens een financieel voorbeeld, waarbij dit goed kan worden toegepast.

5. Denkt u dat de keuze van de startwaarden voor de simulatie er veel toe kunnen doen? Hoe dient men te handelen t.a.v. deze problematiek?

6. Bij een bestaand financierings-support-systeem (FSS) wordt gesteld, dat er sprake is van het gebruik van *twee* modellen, namelijk een simulatie- en een optimaliseringsmodel. De aanvaardbare hypothesen van de simulatie dienen als input voor het optimaliseringsmodel.
   Legt u eens beknopt, doch wel duidelijk en consistent uit, wat van de voorgaande tekst de betekenis en de bedoeling kan zijn. Is overigens - gezien de aard van de problematiek - optimalisatie wel goed mogelijk? Motiveer uw antwoord.

7. Wat betekent de volgende uitspraak voor het *ontwerpen* en het *gebruik* van simulatiemodellen in het thesaurie-systeem:
   "Het management moet de concrete realisaties steeds kennen en de afwijkingen met de projecties kunnen evalueren, zowel ten aanzien van de kasinkomens als van het rendementsniveau."

OPGAVE IV

De bushalte bij een N.S. station is onderwerp van een onderzoek. De wachttijden blijken nogal eens onaanvaardbaar op te lopen. Bovendien is de afstemming op aankomst en vertrek van treinen slecht geregeld. De bushalte in kwestie is geen begin- of eindpunt, doch ligt ergens halverwege in een verkeerstechnisch drukke route. Er blijken veel treinreizigers van de bus gebruik te maken, gezien de vele busreizigers die bij de halte uitstappen (gewoonlijk meer dan de helft van het aantal inzittenden) en ook de grote hoeveelheid reizigers die hier instappen (meestal nog meer dan het aantal uitstappers).

De treinen rijden stipt op het halve uur, bijvoorbeeld X.15 en X.45 uur. De bussen rijden in principe om het kwartier, doch beslist niet stipt (X.00, X.15, X.30, X.45). Als de bus 1 of 2 minuten vóór de treinaankomst arriveert, wacht hij de trein af.

Veronderstel dat er geen overstaptijden zijn. Dus de bus van X.15 is op tijd voor de trein van X.15 zowel voor in- als uit- als overstappers.

De bus telt een maximaal aantal plaatsen (bijvoorbeeld 50 personen), dat niet overschreden wordt.

De gegevens, die u ter completering van belang acht, ook ter beantwoording van de volgende vragen, dient u met vermelding van reden zelf te geven.

Eén zo'n gegeven is bijvoorbeeld de veronderstelling dat niemand gaat lopen of een taxi neemt als een volle bus voor zijn neus wegrijdt, doch blijft wachten op de volgende bus.

1. Maak op grond van bovenstaande gegevens een probleemformulering en een modelbeschrijving, tenminste m.b.v. een schets. Welke variabelen neemt u mee, welke zijn stochastisch; welke verdelingen stelt u voor, enzovoort. Motiveren!

2. Welke uitvoer zoudt u wensen teneinde eventuele maatregelen van de vervoersdirectie te kunnen beoordelen? Wat is het nut van simulatie t.a.v. dit onderzoek?

3. Zoudt u om bovenstaand vraagstuk m.b.v. de computer op te lossen bij voorkeur een simulatietaal gebruiken of zouden de specifieke eigenschappen van zo'n taal hier zo onvoldoende benut worden, dat men met een algemene hogere programmeertaal zou kunnen volstaan?

4. Geef de procedures aan op grond waarvan u de steekproefomvang en/of runlengte en het aantal herhalingen gaat bepalen. Bereken een en ander zo mogelijk.

5. U wilt een tweemaal zo grote nauwkeurigheid in uw response-variabele verkrijgen; hoe verandert u de in vraag 4 voorgestelde getallen?

6. Kan men en is het juist om dit systeem 'leeg' te starten. Wat stelt u voor? Wanneer moet de laatste bus (aan het eind van de dag) rijden?

7. Kan men in het experiment van 'multiple response' spreken? ('Multiple response' betekent dat er meer responsevariabelen bestaan, die tot min of meer tegenstrijdige konklusies zouden kunnen leiden). Zo ja, welke kwantitatieve problemen kunt u dan hier verwachten bij validatie?

8. Zijn er in het beschreven 'systeem' simplificaties, die zodanig de juiste gang van zaken vertroebelen dat simulatie-experimenten in het geheel niet betrouwbaar zijn? Hoe komt u daar achter? Probeer relaties te vinden die mogelijk verfijnd kunnen worden.

OPGAVE V

Ralph H. Sprague bespreekt in een artikel*) een financieel planningsmodel, waarvan hier een korte beschrijving volgt.

Het model is ontwikkeld voor een bank.

De invoer bestaat uit doelstellingsvariabelen, marktvariabelen, omgevingsvariabelen en instrumentele variabelen op het gebied van marketing en financiering, historische gegevens, interne en externe gegevens, enzovoort. Met het model is het mogelijk de verrichtingen van de bank voor de periode van een jaar te simuleren. Dan wordt als uitvoer verkregen: een verzameling financiële grootheden, zoals een balans, liquiditeit, een aantal ratio's, enzovoort.

De uitvoer is een voorspelling van de komende 12 maanden, gespecificeerd per maand.

Managers kunnen op deze uitvoer reageren door interactief met het model te gaan experimenteren, alternatieve beslissingsregels in te voeren, enzovoort om een en ander te kunnen evalueren en te vergelijken met de doelstellingen.

Zij kunnen allerlei vragen stellen als: "wanneer hebben we extra kapitaal nodig", "hoeveel zouden we voor een nieuw fonds moeten betalen" of "hoeveel kost het om een veilig liquiditeitsniveau te handhaven".

---

*) 'System Support for a financial planning model', H. Sprague, National Association of Accounts, New York, 1972.

Het gehele systeem bestaat uit een: gegevensmatrix, de simulator en een 'access-system'.

De gegevensmatrix bevat alle in- en uitvoergegevens.

De matrix bevat o.a. een set historische gegevens van de laatste 24 maanden en een set toekomstige gegevens van 12 maanden. Elke maand wordt het model eenmaal 'standaard gedraaid', dan schuift de 'oudste maand' uit de historische gegevens en de pas verstreken maand komt erbij. De simulator geeft de verwachte verrichtingen van de bank voor de komende 12 maanden.

Een onderdeel is een verzameling statistische routines. Hierin bevindt zich o.a. een tijdreeksanalyseprogramma dat de historische gegevens laat 'passen' op een aantal verschillende (verdelings)-functies (Goodness of fit) en dat de best passende kiest.

Deze automatisch gevormde projectie wordt aan de behandelende manager ter beoordeling gegeven, die desgewenst veranderingen kan aanbrengen.

Vervolgens is er de eigenlijke simulatie.

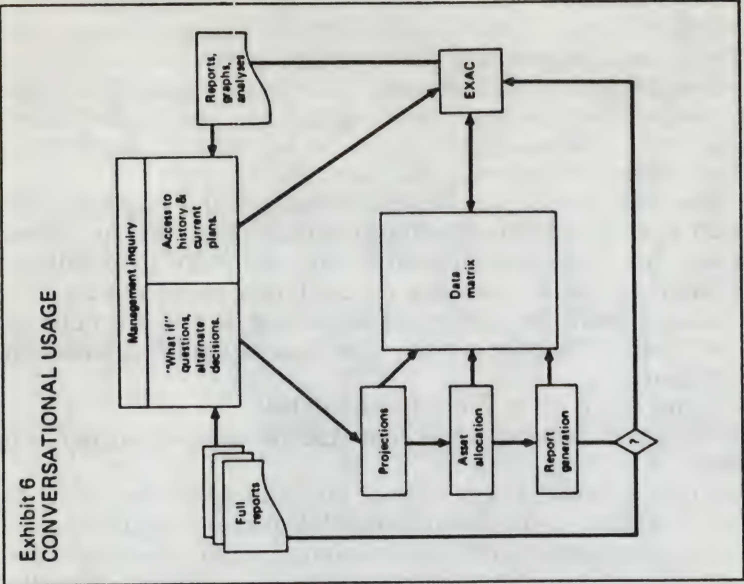
Tenslotte geeft een report generator de uitvoer in een overzichtelijke vorm.

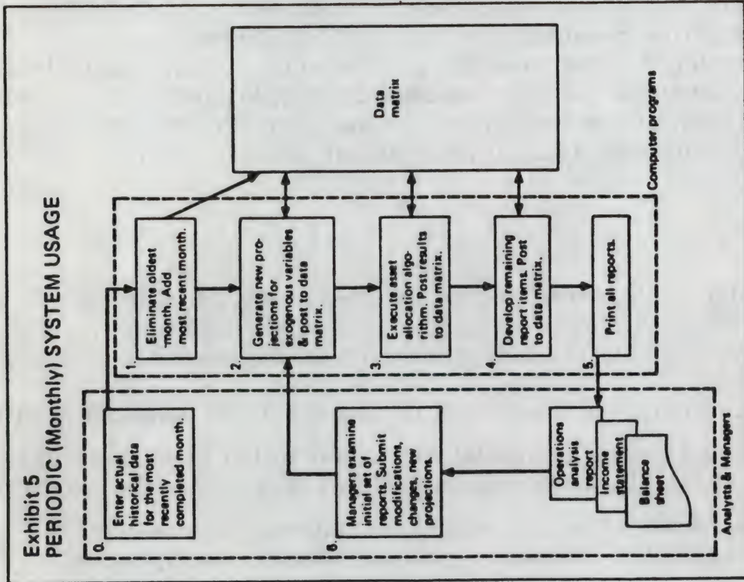
Het 'access-system' zorgt ervoor dat managers en bedrijfsanalisten de database (gegevensmatrix) kunnen benaderen, analyseren en wijzigen, zodat het systeem behalve periodiek (eens per maand) ook gedraaid kan worden wanneer iemand dat wil. Hiervoor is een 'experimentele datamatrix' aanwezig, zodat alle soorten experimenten de maandelijkse run niet verstoren.

Het model is in de praktijk een waardevol en krachtig instrument gebleken, waarmee aan de toenemende complexiteit van de hedendaagse bedrijfsuitvoering tegemoet kan worden gekomen. (Als bijlage ter verduidelijking, treft u een tweetal schema's aan.)

Vragen

1. Stel dat uzelf een opdracht krijgt om een simulatieproject voor een dergelijk financieel planningsmodel uit te voeren. Hoe zou hiervoor simulatie als instrument van onderzoek kunnen worden gebruikt?
   Beantwoord deze vraag aan de hand van de volgende punten:
   - De opzet van het model dat u zou willen gaan gebruiken voor de simulatie-experimenten t.b.v. de onderzoeksfase.
   - De relaties van het eigenlijke ontwerpproces van het beschreven financiële planningsmodel van Sprague.
   - De start van uw onderzoek na het verkrijgen van de opdracht. Hierbij dient tamelijk gedetailleerd aan de orde te komen de
     • oriënterende fase;
     • systeem- en probleemdefinitie;
     • het vinden en ordenen van het waarnemingsmateriaal;
     • het formuleren van hypothesen;
     • enzovoort.


Exhibit 6
CONVERSATIONAL USAGE
Management inquiry
"What if" questions, alternate decisions.
Access to history & current plans.
Reports, graphs, analyses
EXAC
Data matrix
Projections
Asset allocation
Report generation
?
Full reports


Exhibit 5
PERIODIC (Monthly) SYSTEM USAGE
0. Enter actual historical data for the most recently completed month.
1. Eliminate oldest month. Add most recent month.
2. Generate new projections for exogenous variables & post to data matrix.
3. Execute asset allocation algorithm. Post results to data matrix.
4. Develop remaining report items. Post to data matrix.
5. Print all reports.
6. Managers examine initial set of reports. Submit modifications, changes, new projections.
Data matrix
Computer programs
Analysts & Managers
Operations analysis report
Income statement
Balance sheet

2. Stel een conceptueel model op van dat gedeelte uit het planningsmodel dat simulatiekenmerken bevat en geef voor uw model de exogene en endogene variabelen en de verzameling relaties en parameters.

3. Waarom zoudt u het genoemde tijdreeksanalyseprogramma invoeren? Zoudt u ook een programma op willen nemen voor gevoeligheidsanalyse?

4. Probeer een zo goed mogelijke opzet te maken voor *uw* experimenteel ontwerp.

5. Wat is in verband hiermee de doelstelling van *uw* simulatie?

6. Aan welke eisen t.a.v. het gebruikende management zoudt u tegemoet willen komen? Op welke manier?

7. Zou het gebruik van de taal SIMULA op bezwaren van het management kunnen stuiten?

8. Aan welke eisen t.a.v. het programmeren van het model moet worden voldaan?
   Om welke reden zou het in de genoemde modules verdeeld zijn?

9. Wat moet u doen om het management te overtuigen dat
   a. het model betrouwbaar is om voor de bank te nemen beleidsmaatregelen te ondersteunen.
   b. het model zoveel oplevert dat het zijn geld waard is.

10. Vindt u het model zoals dit in de opgave beschreven is wel een 'echt' simulatiemodel?
    Kunt u criteria definiëren op basis waarvan een uitspraak of een willekeurig model een simulatiemodel genoemd mag worden kan worden gedaan?

De volgende opgaven zijn o.a. op het punt van vraagstelling van een andere aard dan de voorgaande. Terwijl de vragen van de eerste vijf opgaven heel concreet zijn, is de vraagstelling nu nogal open. De bedoeling van deze opgaven is dat de studenten ze geheel zelfstandig uitwerken, dus inclusief een computerprogramma. De uitwerking kan als onderdeel van een verslag worden opgenomen.

Gevraagd zou kunnen worden om in dit verslag aandacht te besteden aan de volgende punten:

- probleemstelling;
- formulering doelstelling(en) van het onderzoek;
- formulering van het model;
- het experimenteel ontwerp;
- computerprogramma;
- validatie, waaronder resultatenanalyse.

Zie voor een voorbeeld hoofdstuk IX, de case 'Ontladen'.

OPGAVE VI

Simulatie van een centrale meldpost*)

In één van onze naburige landen is een groot coöperatief bankconcern bekend onder de naam: de 'Verenigde Vlaamse Gemeenten' ofwel VVG.

In alle plaatsen van enige betekenis bevinden zich één of meer lokale banken, die bij de VVG zijn aangesloten.

Voor de afhandeling van de transacties die zich bij de banken afspelen beschikken de plaatselijke banken over computersystemen. Bij de plaatselijke banken zijn op dit moment 260 computers geïnstalleerd van type B34 en 190 van type NX8.

Over niet al te lange tijd wordt dit aantal verder uitgebreid. Men verwacht dat over een half jaar het aantal systemen van B34 310 zal bedragen en van type NX8 220.

Wanneer er storingen bij de computersystemen optreden, dan worden deze telefonisch doorgegeven aan het hoofdkantoor van de VVG. Daar bevindt zich een centrale meldpost waar een staf van deskundigen met specialistische kennis over de typen B34 en NX8 de telefonische storingsmeldingen opvangen.

In vrijwel alle gevallen kunnen de problemen door middel van het telefoongesprek worden opgelost.

Er zijn twee soorten storingsmeldingen, namelijk: technische problemen en bedieningsproblemen.

De duur van het telefoongesprek is afhankelijk van zowel het soort computersysteem als ook van het soort probleem.

---

*) Met dank aan dr.ir. J. Koene.

De verdeling van de duur van de gesprekken is in de volgende tabel weergegeven:

Duur in minuten	type B34 technische problemen	type B34 bedienings-problemen	type NX8 technische problemen	type NX8 bedienings-problemen
0-2	0.0	10.4	0.0	6.1
2-5	5.2	62.3	1.7	17.8
5-10	9.6	20.7	4.8	54.3
10-15	39.0	3.6	26.9	17.6
15-20	27.8	1.7	50.1	2.1
20-25	15.7	0.9	14.7	1.2
25-30	2.7	0.4	1.8	0.9

Het gemiddeld aantal telefoongesprekken per week per 100 systemen van type B34 bedraagt 55.6.
Voor type NX8 is dit aantal 67.7.
Van alle meldingen voor type B34 is 60% een technisch probleem.
Bij type NX8 is slechts 40% van technische aard.
Maandag blijkt de drukke dag: het aantal telefoontjes is 20% hoger dan op de andere werkdagen.

Een werkdag is verdeeld in 4 gelijke perioden van 2.5 uur. De werkdag duurt van 8.00-18.00 uur. De eerste periode (van 8.00-10.30 uur) geeft 50% meer telefoongesprekken dan in de andere perioden.

De leiding van de meldpost in het hoofdkantoor van de VVG wil inzicht krijgen in de gewenste hoeveelheid personeelsleden (eventueel zou kunnen worden overwogen om de specialistische kennis te splitsen naar de typen computersystemen) en in het aantal benodigde telefoonlijnen over een half jaar.

Uiteraard spelen factoren als service, kosten en bezettingsgraad een grote rol.

Om aan de wensen van de leiding tegemoet te komen wordt een simulatiemodel gemaakt van de centrale meldpost.

Er wordt u gevraagd om dit model te ontwikkelen.

OPGAVE VII

De brugwachter

Een brugwachter zit met het probleem dat hij zowel automobilisten als watersporters te vriend moet houden. Aangezien het bij de brug zowel op het water als op de weg erg druk is, ontstonden er

in het verleden veel problemen. De brugwachter verzoekt u voor hem een beslissingsregel te formuleren, waaraan hij zich bij het openen van de brug kan houden.

De brugwachter is van mening, dat het autoverkeer voorrang heeft maar dat de gemiddelde wachttijd voor boten toch niet groter mag zijn dan 5 minuten.

Over de situatie bij de brug zijn de volgende gegevens bekend:

- *op de weg:* De aankomst van auto's is te beschrijven met een negatief exponentiële verdeling. Van beide kanten komen er 10 auto's per $2\frac{1}{2}$ minuut. Als er geen file is kan de tijd die de auto's nodig hebben om de brug over te steken verwaarloosd worden.
  Wanneer er een file is ontstaan (door het openen van de brug) kan gesteld worden dat de auto's van beide kanten, één voor één over de brug rijden. De auto's vertrekken dan om de 3 seconden.
- *op het water:* De aankomst van de boten is negatief exponentieel verdeeld met (van beide kanten) een gemiddelde van 20 boten per uur.
- *de brug:* Als de brug geopend wordt duurt dit 60 seconden. Hierna kunnen de boten van één kant tegelijk één voor één onder de brug doorvaren. De doorvaartijd is uniform verdeeld op het interval 15-35 seconden. Wanneer de brug gesloten wordt duurt dit wederom 60 seconden.

Onderzoek het gedrag van het systeem en adviseer de brugwachter omtrent een te volgen beslissingsregel t.a.v. het openen van de brug.

Opmerking:

Mogelijke regels zijn bijvoorbeeld:
1. doe de brug om de x minuten y minuten open;
2. doe bij x boten de brug open tot alle boten erdoor zijn;
3. doe bij x boten de brug y minuten open.

OPGAVE VIII

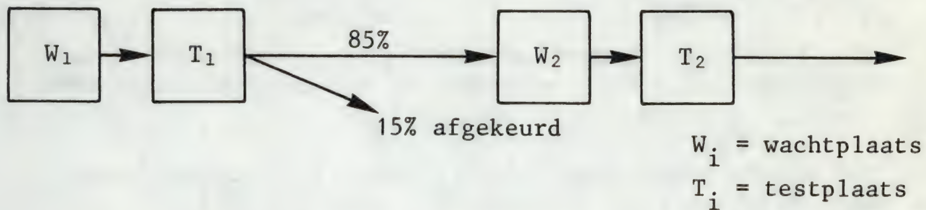
Autokeuring

Een vereniging van automobilisten is bezig met het ontwerp van een keuringsstation voor auto's. Dit keuringsstation gaat bestaan uit twee testapparaten. Als het station in werking is gesteld kan de leiding de auto's op afspraak laten komen. Het keuringsstation

gaat bestaan uit één wachtplaats voor één auto, gevolgd door het eerste testapparaat, daarna een tweede wachtplaats voor één auto en tenslotte het tweede testapparaat.

De helptijd op het eerste testapparaat is normaal verdeeld met gemiddeld drie minuten en een standaarddeviatie van ½ minuut. Na de eerste test is 15% van de auto's afgekeurd. Deze verlaten het station. De overige auto's worden ook op het tweede apparaat getest. De testtijd is hier ook normaal verdeeld met een gemiddelde van vier minuten en een standaarddeviatie van één minuut. Buiten het station is geen wachtruimte voor auto's.

Deze opstelling is schematisch als volgt weer te geven:



De gevolgen van deze opstellingen zijn (o.a.):

- wanneer er een nieuwe auto komt en W_1 is nog bezet dan moet voor deze auto een nieuwe afspraak worden gemaakt;
- wanneer T_2 en W_2 bezet zijn terwijl de test in T_1 afgelopen is en de auto in T_1 niet afgekeurd is, zal T_1 bezet blijven tot de auto 'door kan schuiven' naar W_2;
- als W_2 leeg is terwijl de test in T_2 is afgelopen, zal T_2 enige tijd ongebruikt zijn.

De leiding kan zelf het aankomstproces bepalen. Zij kan haar afspraken maken om de x minuten. Aangenomen mag worden dat dit aankomstpatroon precies gevolgd zal worden. De leiding van het station wil graag inzicht hebben in de vraag met welke frequentie de auto's bij het station dienen aan te komen.

Uiteraard is voor de directie de bezettingsgraad van het station van belang.

Aangezien de organisatie veel belang hecht aan service wil zij ook het eventuele terugsturen van auto's in haar overwegingen betrekken.

Onderzoek m.b.v. simulatie het gedrag van het systeem. Richt uw onderzoek zodanig in dat de directie inzicht krijgt in de samenhang van de voor haar belangrijke criteria.

OPGAVE IX

Bakkerijsimulatie

Een bakkerij produceert een assortiment gebak dat geleverd wordt aan (door)verkooppunten.

Een verkooppunt bestelt 2 à 3 keer per week.

Het aantal is afhankelijk van de lokale vraag. (Er zijn dag- en seizoensverschillen.)

Tot voor kort leverde de bakkerij op bestelling.

Nu wil de leiding een voorraad aanleggen, want er zijn problemen:

- technische problemen: afstemming van produktie en vraag;
- problemen t.a.v. de voorspelling: de vraag is stochastisch en erg variabel.

Men wenst te vermijden dat er frequent omgeschakeld moet worden van de ene produktieserie gebak naar de andere.

Op vrijdag wordt de voorraad opgenomen en verminderd met de leveringen voor die dag, daarna wordt de produktieplanning voor de volgende week opgesteld.

Voor elke gebaksoort wordt een bepaald aantal produktie-uren beschikbaar gesteld (in principe wordt per gebaksoort één vaste produktiedag vastgesteld).

Extra benodigde capaciteit is eventueel elders in de week beschikbaar.

Ontwikkel een simulatiemodel waarmee het opstellen van het produktieplan wordt ondersteund.

Houd rekening met:

- de servicegraad (neen-verkoop);
- de kwaliteit (maximale houdbaarheid is 2 weken in diepvries);
- het aantal afleveringen per week.

Het simulatiemodel moet inzicht verschaffen in:

- de wekelijks te produceren hoeveelheid;
- voorraad (hoogte en verkoop);
- neen-verkoop;
- te vernietigen voorraden;
- de versheid van het gebak;
- de invloed van wijzigingen in: servicegraad, aantal leveringen per week, beschikbare produktie-capaciteit c.q. produktieplanning.

N.B.

- de bestellingen zijn normaal verdeeld;
- kosten worden in deze simulatie NIET meegenomen.

OPGAVE X

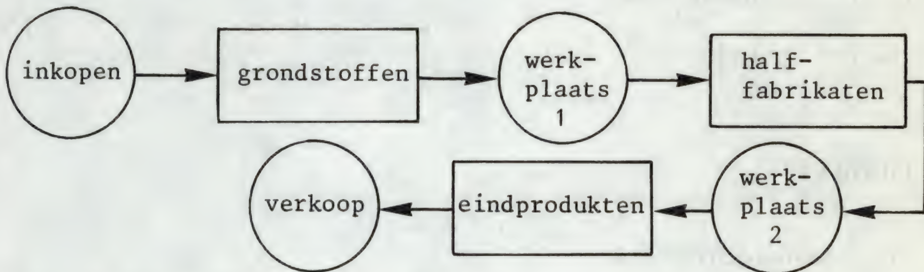
Management game: produktie-simulatie

Ontwikkel een management game, waarbij een aantal managementteams beslissingen moet nemen, die betrekking hebben op het logistieke proces van een produktiebedrijf.

Er moeten beslissingen worden genomen m.b.t. inkoop, produktie en verkoop.

Een belangrijk aspect van een management game is het competitie-element. Het spel moet de teams zodanig motiveren dat ze al hun kennis en inzicht zullen aanwenden om zo goed mogelijke beslissingen te nemen.

Als uitgangspunt nemen we het volgende logistieke systeem:



Er is een markt waar de grondstoffen gekocht worden, deze kent uiteraard een maximum aanbod en er is een markt waar de eindprodukten worden afgezet. Deze laatste kent een totale vraag voor de bedrijven (managementteams) die hier op opereren.

Van belang is dat de wezenlijke zaken uit de werkelijkheid in het model/spel worden opgenomen. Het model dient aan de andere kant eenvoudig te zijn zodat de verbanden eenvoudig te herkennen zijn.

Een aantal aspecten en elementen die onder andere in het model opgenomen dient te worden:

- één type eindprodukt;
- inkoop- en verkoopmarkt met respectievelijk een totaal aanbod en een totale vraag;
- een vraag naar het eindprodukt per periode die een stochastisch element bevat en tevens een trend en/of conjunctuur kan volgen, met parameters die door de spelleider ingevuld c.q. aangepast kunnen worden;
- analoog voor het aanbod van de grondstoffen;
- de verkoopprijs is voor alle bedrijven dezelfde, evenals de inkoopprijs;
- er is een levertijd voor de grondstoffen;
- per produktieproces is er uitval (stochastisch);

- de magazijn(opslag)capaciteit en de produktiecapaciteit is aan te passen;
- er zijn produktiekosten, magazijnkosten en bestelkosten;
- neen-verkopen zullen de vraag bij dat bedrijf in enkele daaropvolgende perioden negatief beïnvloeden;
- het aanbod op de grondstoffenmarkt ijlt na op de totale vraag naar eindprodukten hetgeen de inkoopprijzen bij krapte in aanbod zal opdrijven en omgekeerd zal doen dalen bij enorm overaanbod.

Van belang bij het spelen is een snelle verwerking van de beslissingen, overzichtelijke presentatie van de resultaten van iedere ronde en het gemak en het aantal processen dat de spelleider kan beïnvloeden (op grond van het spelverloop). Een overzicht van het totaal, naast de overzichten voor de afzonderlijke managementteams, is dus vereist.

Men dient ook een (korte) handleiding te schrijven voor degenen die het spel spelen.

OPGAVE XI

Datacommunicatienetwerk

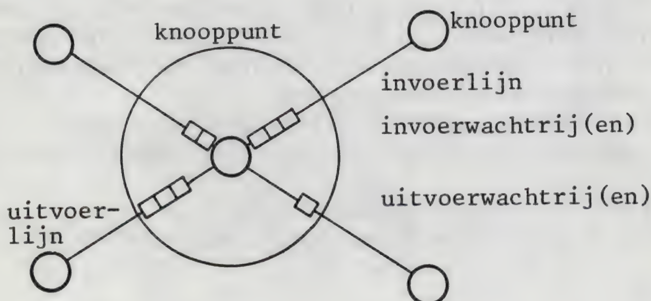
In een automatiseringscentrum is een datacommunicatienetwerk geïnstalleerd, waarbij gebruik wordt gemaakt van twee verstuuralgoritmen van 'packet-switching', namelijk de Hot-Potato Methode (HPM) en Static Routing (SR). Er wordt aangenomen dat er geen storingen in het datacommunicatienetwerk kunnen optreden.

Toelichting

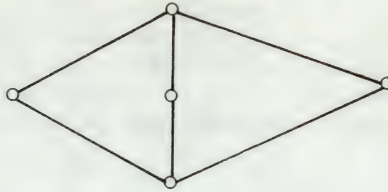
Netwerk:

Onder een datacommunicatienetwerk wordt verstaan een systeem, dat bestaat uit knooppunten en verbindingen daartussen (lijnen).

Een knooppunt bestaat uit een processor, waar berichten gegenereerd kunnen worden en waarvandaan pakketjes doorgestuurd kunnen worden.



Een voorbeeld van een netwerk is in de volgende schets weergegeven:



Packet switching:

Een bericht dat door het netwerk verstuurd moet worden, wordt opgedeeld in pakketjes van een vaste grootte (bijvoorbeeld pakketjes van 512 bytes), die onafhankelijk over het netwerk worden verstuurd. Een software-pakket ('Packet Assembling and Disassembling') draagt zorg voor het koppelen van de pakketjes tot het bericht op het knooppunt van bestemming.

Verstuuralgoritme:

Een verstuurmethode zorgt ervoor dat pakketjes op hun knooppunt van bestemming aankomen.

Hot-Potato Methode (HPM):

Op het moment dat een pakketje ter verzending wordt aangeboden, wordt door het algoritme de kortste wachtrij van de lijnen (uitgezonderd de wachtrij van de lijn, met als eindpunt het knooppunt waarvandaan het pakketje het laatst verzonden is) bepaald.
De wachtrij van de lijn met de kortste wachtrij zal verlengd worden met het te verzenden pakketje.

Static Routing (SR):

Op het moment dat een pakketje ter verzending wordt aangeboden, bepaalt het algoritme de lijn waarover het pakketje verzonden moet worden en plaatst het in de wachtrij van deze lijn.
De wijze waarop de lijn gekozen wordt is als volgt:

> De lijnen zijn volgens een kansverdeling (deze kansverdeling is afhankelijk van de lijn waarop het pakketje binnenkomt òf van het feit dat het pakketje op het betreffende knooppunt gegenereerd is) ingedeeld. Hierbij kan een pakketje niet naar het knooppunt terugkeren, waarvandaan het zoëven gekomen is.
> Aan de hand van een gegeven kansverdeling (een onderverdeling in intervallen) wordt een lijn gekozen.
> Daarna wordt het pakketje achteraan in de juiste wachtrij gezet.

HPM en SR zijn twee algoritmen met pakketversturing, maar er bestaan meer algoritmen. De keuze valt op HPM en SR, omdat zij beide een groep algoritmen vertegenwoordigen, namelijk respectievelijk gedecentraliseerde en gecentraliseerde algoritmen.

Gedecentraliseerd wil zeggen dat alleen naar de status van het knooppunt gekeken wordt, namelijk de lijn met de kortste wachtrij.

Gecentraliseerde algoritmen daarentegen zoeken een optimale situatie voor het gehele netwerk, aan de hand van waarschijnlijkheidstabellen.

Vergelijk m.b.v. simulatie de twee verstuurmethoden HPM en SR onder verschillende omstandigheden binnen het netwerk.

APPENDIX A

Statistische Technieken

A1 INLEIDING

Indien simulatie als methode van onderzoek wordt toegepast, is het is het gebruik van statistische technieken noodzakelijk. Zowel bij modelbouw als bij experimenteel ontwerp en validatie wordt van statistiek gebruik gemaakt. In deze appendix wordt een aantal van de meest gebruikte technieken uit de statistiek in het kort beschreven.

De beschrijvingen zijn zo summier, dat ze waarschijnlijk slechts begrepen en gebruikt kunnen worden door hen die in hun opleiding één of meer cursussen statistiek hebben gevolgd. De appendix is als 'geheugenopfrisser' voor deze groep bedoeld.

Zij die simulatie als onderzoekmethode willen gebruiken, maar nog niet over de vereiste statistische kennis beschikken raden we aan een goed statistisch leerboek*) door te nemen, alvorens te gaan experimenteren. Daarnaast is het aan te bevelen, zeker als men over weinig ervaring beschikt, een statistisch expert te raadplegen.

Wij wijzen met nadruk op de mogelijkheden die een statistisch softwarepakket (bijvoorbeeld SPSS, BMDP of WESP) biedt. De voordelen van het gebruik van een dergelijk pakket wegen ruimschoots op tegen de investeringen van geld en tijd. Wel is ook in dit verband een waarschuwing op zijn plaats: gebruik softwarepakketten nooit als 'black-boxes'. Verdiep u eerst in de inhoud, in de mogelijkheden van zo'n pakket alvorens er gebruik van te maken voor simulatietoepassingen.

---

*) Zie bijvoorbeeld: Mark L. Berenson en David M. Levine, Basic Business Statistics, Sec. edition, Prentice Hall, 1983.

A2 SCHATTING VAN PARAMETERS

Laten we aannemen dat we waarnemingen doen voor een stochastische variabele $\underline{X}$ (stochastische variabelen worden onderstreept). Dan noemen we de waarneming x een steekproef van $\underline{X}$. $\underline{X}$ bezit een verdelingsfunctie F(X), evenals een verwachtingswaarde μ en een variantie σ^2.

De waarnemingen x_1, x_2, x_3, ... zijn steekproeven uit respectievelijk $\underline{X}_1$, $\underline{X}_2$, $\underline{X}_3$, ... met verschillende verdelingsfuncties $F(X_1)$, $F(X_2)$, $F(X_3)$, ...

In een loket situatie is bijvoorbeeld de rijlengte na het begin van de simulatie (mits het systeem 'leeg' gestart is) waarschijnlijk geringer, dan wanneer na enige tijd een aantal klanten met korte tussenpozen is binnengekomen. Is op een bepaald tijdstip de rijlengte groot, dan zal dit op de momenten vlak daarna niet veel anders zijn.

Vaak kan aangenomen worden dat de verdelingsfuncties naar een bepaalde grenswaarde convergeren. Dit betekent dat voor een voldoende grote n alle variabelen $\underline{X}_n$ dezelfde verdelingsfunctie hebben, F(X).

De waarnemingenreeks zal dan, als tevens aan enkele andere voorwaarden is voldaan, *stationair* genoemd worden. Bij het loket mag men aannemen, dat, als de bedieningstijden gemiddeld kleiner zijn dan de tussenaankomsttijden, na een zekere beginperiode (Engels: transient phase) een stationaire toestand ontstaat.

Is daarentegen de verhouding bedieningstijd/aankomsttijd ongunstig, dan groeit de wachtrij steeds aan: het systeem explodeert en de stationaire toestand wordt nooit bereikt. De grens voor een te druk bezet systeem ligt ongeveer bij een bezettingsgraad van 0.8.

Van de uitvoervariabele X, bijvoorbeeld wachttijd, zou men graag de verdelingsfunctie willen weten. Daartoe doet men een steekproef van n herhaalde waarnemingen $\underline{X}_1$, $\underline{X}_2$, $\underline{X}_3$, ..., $\underline{X}_n$.

Vaak is het voldoende om van die verdelingsfuncties de verwachtingswaarde μ en de variantie σ^2 te kennen. Hiervoor bestaan eenvoudige schattingsfuncties.

a. Enkelvoudige steekproeven

Een *steekproef* uit een eindige populatie is een deelverzameling van die populatie. Bij een *kanssteekproef* (Engels: probability sample) kan voor ieder element van de populatie bepaald worden hoe groot de kans is dat dit element tot deze steekproef behoort. Deze kansen zijn meestal gelijk, soms evenwel ongelijk.

De *enkelvoudige* (aselecte) steekproef (Engels: simple random sample) bezit de eigenschap dat alle steekproeven van dezelfde omvang dezelfde trekkingskans bezitten. Alle elementen van de

populatie hebben dan even grote trekkingskansen. Onder een *aselecte* steekproef (Engels: random sample) wordt een enkelvoudige steekproef verstaan, meestal met teruglegging (onafhankelijke trekkingen).

ASELECTE		NIET-ASELECTE
ENKELVOUDIGE	NIET-ENKELVOUDIGE	
a) Zonder teruglegging b) Met teruglegging	a) (aselecte) gelede steekproeven b) (idem) trossteekproeven c) (idem) getrapte steekproeven d) systematische steekproeven e) diverse steekproeven	a) Uit alle elementen van de populatie b) Uit de trossen

Tabel A.1 Soorten van steekproeven

Zie voor een uitgebreide behandeling van de in Tabel A.1 opgesomde soorten steekproeven, bijvoorbeeld [1] of [2].

b. *Schatting bij enkelvoudige steekproeven*

Schatten van een onbekende parameter van een populatie is het berekenen op grond van een kanssteekproef. Men onderscheidt *schatters* (methoden of formules) en *schattingen* (uitkomsten daarvan). Belangrijk zijn de *zuiverheid* en de *doeltreffendheid* van schatters.

Bij een enkelvoudige steekproef berekent men het steekproefgemiddelde $\bar{x}$; dat is een zuivere schatter van μ, het populatiegemiddelde.

In formulevorm:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$ N is de populatie-omvang

$$\underline{\bar{x}} = \frac{1}{n} \sum_{i=1}^{n} \underline{x}_i$$ n is de steekproefomvang

De variantie in de steekproef wordt gedefinieerd als:

$$\underline{s}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\underline{x}_i - \overline{\underline{x}})^2 = \frac{1}{n-1} \sum_{i=1}^{n} \underline{x}_i^2 - n \cdot \overline{\underline{x}}^2 \quad .$$

Dit is een zuivere schatter van

$$\sigma^2 = \frac{1}{N} \Sigma (x_i - \mu)^2 \quad .$$

De verwachting $E\overline{\underline{x}}$ is het gemiddelde van de steekproefgemiddelden in de steekproefruimte. De variantie van de steekproefgemiddelden wordt gegeven door:

$$\sigma^2(\overline{\underline{x}}) = \frac{\sigma^2}{n}$$

bij trekking met teruglegging of indien $N \gg n$.

Deze grootheid is te schatten d.m.v.:

$$\underline{s}^2_{\overline{\underline{x}}} = \frac{\underline{s}^2}{n}$$

De spreiding $\frac{\underline{s}}{\sqrt{n}}$ dient als schatting van de fout van $\overline{\underline{x}}$ t.o.v. μ.

Men lette goed op deze grootheid. Immers de fout in omgekeerd evenredig met $\sqrt{n}$, waaruit blijkt dat als men een tweemaal zo kleine fout wenst daarvoor viermaal zoveel waarnemingen nodig zijn.

Als de voorwaarden van teruglegging of $N \gg n$ niet gelden dan wordt de variantie als volgt gegeven:

$$\sigma^2(\overline{\underline{x}}) = \frac{\sigma^2}{n} \left(1 - \frac{n}{N}\right) \quad .$$

De wortel uit de variantie van het steekproefgemiddelde heet de *standaardfout* (Engels: standard error) van dit gemiddelde (Engels: sample mean).

De variantie van $\overline{\underline{x}}$ wordt zuiver geschat door middel van:

$$\underline{S}^2 = \text{var } \underline{x} = s^2_{\overline{\underline{x}}} = \frac{s^2}{n} \left(1 - \frac{n}{N}\right) \quad .$$

Met behulp van $\overline{\underline{x}}$ berekent men *betrouwbaarheidsgrenzen* en *betrouwbaarheidsintervallen*, mits n groot is. Men spreekt van intervalschatting. Het 95%-betrouwbaarheidsinterval voor het gemiddelde μ bij een steekproef uit een normale verdeling met onbekende variantie is, mits n voldoende groot is:

$$\underline{\bar{x}} - 1{,}96\,\frac{\underline{s}}{\sqrt{n}} < \mu < \underline{\bar{x}} + 1{,}96\,\frac{\underline{s}}{\sqrt{n}}$$

Voor kleine steekproefomvang moet in dit geval de $\underline{t}$-verdeling worden toegepast ($n < 30$).

De $\underline{t}$-verdeling van *Student* is van belang als men een steekproef uit een normale verdeling heeft waarvan σ onbekend is.
Dan bezit

$$\frac{\underline{\bar{x}} - \mu}{\sigma/\sqrt{n}}$$ een N(0,1)-verdeling en

$$\frac{\underline{\bar{x}} - \mu}{\underline{s}/\sqrt{n}}$$ een $\underline{t}$-verdeling met (n-1) vrijheidsgraden.

c. *Het betrouwbaarheidsinterval*

$$\underline{\bar{x}} - t_{n-1}\,\frac{\underline{s}}{\sqrt{n}} < \mu < \underline{\bar{x}} + t_{n-1}\,\frac{\underline{s}}{\sqrt{n}}$$

kan voor verschillende overschrijdingskansen berekend worden.

Als σ onbekend is terwijl wel aan de eis van de normale verdeling is voldaan, dan zijn de grenzen van betrouwbaarheidsintervallen voor μ:

$$\underline{t}_1 = \underline{\bar{x}} + \frac{t_{\frac{1}{2}\alpha} \cdot \underline{s}}{\sqrt{n}} \qquad \text{en} \qquad \underline{t}_2 = \underline{\bar{x}} - \frac{t_{1-\frac{1}{2}\alpha} \cdot \underline{s}}{\sqrt{n}}$$

waarin $t_{\frac{1}{2}\alpha}$ en $t_{1-\frac{1}{2}\alpha}$ moeten worden afgelezen uit een tabel voor de gekozen α en (n-1) graden van vrijheid.

Bijvoorbeeld: $\bar{x} = 3,\ s = 1,\ n = 20,\ \alpha = 0{,}05$

dan is: $t_{\frac{1}{2}\alpha} = t_{1-\frac{1}{2}\alpha} = t_{0,975} = 2{,}093$

$$t_1 = 3 + \frac{2{,}09}{\sqrt{20}} = 3{,}47 \qquad t_2 = 3 - \frac{2{,}09}{\sqrt{20}} = 2{,}53$$

In 95% van de op deze manier m.b.v. de (steeds verschillende) steekproefgemiddelden te berekenen intervallen zal μ binnen de grenzen van het interval liggen.

Anders gezegd: het interval I is $2{,}53 \leq \mu \leq 3{,}47$; I is één van de 95 van de 100 intervallen waarin μ zal liggen.

Omdat voor grotere steekproeven, vanaf ongeveer $n = 30$ de t-tabel overgaat in de tabel van de normale verdeling kan voor deze steekproef de tabel van de normale verdeling worden gebruikt.

d. De steekproefgrootte

Bij een 95%-betrouwbaarheidsinterval en een gegeven nauwkeurigheidspercentage b kan de steekproefgrootte n worden berekend uit de formule:

$$n = \left(\frac{1{,}96 \cdot \underline{S}}{b \cdot \bar{x}}\right)^2$$

Als de veronderstelling van een normale verdeling niet kan worden gemaakt, kan voor de *bepaling* van de steekproefomvang de regel van Tsjebisjef (Chebycheff) worden toegepast. Deze regel luidt:

$$P\left(|\underline{\bar{x}} - \mu| > k \cdot \frac{\sigma}{\sqrt{n}}\right) \leq \frac{1}{k^2}$$

De kans dat, ongeacht de verdeling het verschil tussen het steekproefgemiddelde en het populatiegemiddelde groter is dan $k \cdot \frac{\sigma}{\sqrt{n}}$ is hooguit gelijk aan $\frac{1}{k^2}$. De regel zegt bijvoorbeeld dat buiten het interval

$$\mu + 2 \cdot \frac{\sigma}{\sqrt{n}} \quad \text{en} \quad \mu - 2 \cdot \frac{\sigma}{\sqrt{n}}$$

hoogstens 1/4 der waarnemingen ligt. Dit is dus nogal wat ruimer dan bij toepassing van de normale verdeling.

In de volgende tabel is een aantal waarden gegeven voor:

a. toepassing van de normale verdeling

b. de regel van Tsjebisjef, elk bij:
   - verschillende waarden van d, d.i. de afwijking van $\underline{x}$ t.o.v. μ
   - onbetrouwbaarheidsdrempel $\alpha = 0{,}05 = \frac{1}{k^2}$.

d	a	b
σ/2	15	80
σ/4	61	320
σ/6	128	720
σ/8	246	1280
σ/20	1537	8000

Tabel A.2 Steekproefomvang bij toepassing van normale verdeling en regel van Tsjebisjef

A3 TOETSING

a. Het toetsen van een hypothese

Een statistische hypothese is een uitspraak over de populatie, waaruit een steekproef is getrokken.

Deze uitspraak kan worden getoetst op haar geldigheid door een statistische toetsingsmethode. Dit is een verzameling regels, waarmee een hypothese wel of niet wordt geaccepteerd. Bij een dergelijke beslissing ontstaan twee typen foutmogelijkheden.

De uitspraak welke men wenst te toetsen heet de nul-hypothese H_0 terwijl men met H_1 een alternatieve uitspraak aanduidt.

Bijvoorbeeld: H_0: $\mu = \mu_0$

en H_1: $\mu \neq \mu_0$.

Als men H_0 verwerpt, terwijl H_0 juist is, maakt men een 'type-I-fout' of 'α-fout'.

Het is natuurlijk ook mogelijk dat men H_0 ten onrechte accepteert. Dan maakt men een 'type-II-fout' of een 'β-fout'.

Toetsingsresultaat	werkelijkheid	
	H_0 juist	H_0 onjuist
H_0 geaccepteerd	correct	β-fout
H_0 verworpen	α-fout	correct

Uiteraard wenst men onjuiste beslissingen te voorkomen; daartoe gebruikt men statistische toetsen om de kans op α en β fouten zo klein mogelijk te maken.

Een toets bestaat uit een zestal opeenvolgende stappen.

1. Stel op grond van praktische overwegingen vast, welke hypothese getoetst moet worden (bepaal H_0 en H_1).

2. Bepaal het aanvaardbare risico α dat een geldige nul-hypothese verworpen wordt (vaak wordt $\alpha = 0.05$ of $\alpha = 0.01$ gekozen).

3. Kies een geschikte toetsingsmethode om H_0 te toetsen.

4. Stel de steekproefverzameling vast.

5. Ga na hoe het kritieke gebied, waarin H_0 verworpen wordt, er uitziet.

6. Neem een steekproef van vooraf bepaalde grootte (n), bereken de statistische formules en trek conclusies over de hypothese.

Een eenvoudig voorbeeld

De uitspraak omtrent de aankomst bij een loketsysteem is dat gemiddeld 10 klanten per tijdseenheid arriveren.

De nul-hypothese luidt:

$$H_0 : \mu = 10$$

$$H_1 : \mu \neq 10 .$$

Stel $\alpha = 0.05$ (5% risico).
Men test het gemiddelde. Daarvoor geldt het volgende:

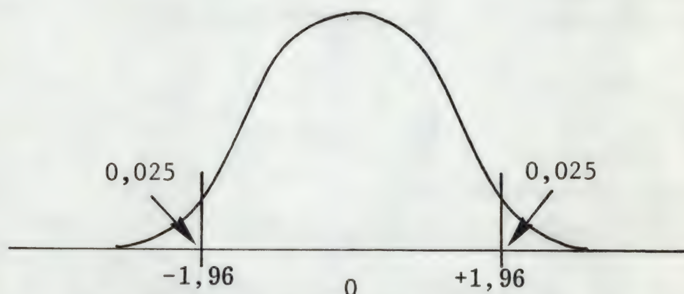
$$\underline{U} = \frac{\bar{\underline{x}} - \mu}{\sigma/\sqrt{n}}$$

(Voor 'U' vindt men in de literatuur ook vaak 'Z'.)

$\bar{\underline{x}}$ is normaal verdeeld met gemiddelde μ en standaardafwijking $\sigma/\sqrt{n}$. (Dit geldt alleen als n groot genoeg is, volgens de centrale limietstelling.)

H_0 wordt verworpen als

$$\left|\frac{\bar{x} - \mu}{\sigma/\sqrt{n}}\right| > 1.96 .$$



Figuur A.1 Het kritieke gebied

Stel nu, dat de steekproefomvang 16 bedraagt en het gemiddelde $\bar{x} = 12$. Verder blijkt $\sigma = 2$.

Dan geldt:

$$U = \frac{12 - 10}{2/4} = 4$$

H_0 wordt verworpen en de conclusie luidt dat $\mu \neq 10$ bij $\alpha = 0,05$.

Wanneer σ^2 onbekend is, moet men σ vervangen door s, terwijl in plaats van de normale verdeling, de t-verdeling moet worden gebruikt.

Vaak komt het voor dat twee alternatieven vergeleken moeten worden. Dan kan de hypothese luiden dat beide populaties hetzelfde gemiddelde hebben.

Veronderstel hiervoor twee normaal verdeelde populaties, met gemiddelde μ_1 en variantie σ_1^2 respectievelijk μ_2 en σ_2^2.

$$H_0: \mu_1 = \mu_2 \qquad \text{en} \qquad H_0: \mu_1 - \mu_2 \leqq \varepsilon \ .$$

Zijn de populatievarianties bekend, dan wordt de U-verdeling gebruikt; in het andere geval de t-verdeling.

Men kan ook geïnteresseerd zijn in de varianties, in plaats van in de gemiddelden. Wanneer men als hypothese gelijkheid van variantie van twee populaties wil veronderstellen (H_0: $\sigma_1^2 = \sigma_2^2$), dan wordt de F-toets gebruikt.

b. *'Goodness of Fit'*

Een veel voorkomende situatie bij simulatie is, dat men van een verzameling stochastische variabelen wenst te weten of deze verzameling 'past' bij een bekende theoretische verdeling.

Om dit te meten bestaan toetsen, waarvan de Chi-kwadraat-toets en de Kolmogorov-Smirnov-toets bekende en veel gebruikte hulpmiddelen zijn.

De chi-kwadraat verdeling wordt gegeven door:

$$\chi^2 = \sum^{n} \frac{(f_0 - f_e)^2}{f_e}$$

f_0 is de waargenomen frequentie;

f_e is de theoretisch berekende frequentie.

Wanneer de berekende χ^2 groter is dan de theoretische χ^2, die men in tabellen kan opzoeken, verwerpt men de hypothese (H_0) dat de waargenomen en de theoretische verdeling dezelfde zijn.

Een voorbeeld

Aan een tentamen organisatiekunde is deelgenomen door 154 studenten. De indruk bestaat dat de tentamenresultaten normaal verdeeld zijn. De gemiddelde puntenscore is 73,3 en de standaardafwijking 10,05.

De nulhypothese is in dit voorbeeld: de verdeling van de tentamenresultaten is normaal met een gemiddelde van 73,3 en een standaardafwijking van 10,05.

In de volgende tabel zijn de resultaten in klassen weergegeven met de bijbehorende berekeningen ten behoeve van de chi-kwadraattoets.

puntenscore	f_0 (waargenomen)	f_e (theoretisch)	$\frac{(f_0 - f_e)^2}{f_e}$
< 59,5	16	13,1	0,64
60 - 69	31	41,3	2,57
70 - 79	72	58,1	3,33
80 - 89	26	33,3	1,60
≧ 89,5	9	8,3	0,06
	154	151,9	8,20

Het aantal graden van vrijheid is n-p-1 = 5-2-1 = 2. In de tabel van de χ^2-verdeling vinden we $\chi^2_{0.99,2}$ = 9,21. Daar 8,20 < 9,21 is er geen reden om H_0 te verwerpen.

Een voorwaarde voor toepassing van de χ^2-verdeling is, dat geen der theoretische frequenties kleiner is dan 5 (anders is de benadering onvoldoende). Als aan deze eis niet is voldaan, dan kan worden overwogen bepaalde klassen te combineren.

Als een combinatie van klassen niet de vereiste oplossing geeft of als het aantal klassen te klein wordt of als de steekproef te klein is ($10 \leqq n < 100$) dan is de Kolmogorov-Smirnov-toets een goed alternatief. Men berekent bij deze toets het absolute verschil tussen de cumulatieve waarden van de frequentiegroepen.

De absoluut grootste van deze verschillen wordt vergeleken met een kritische waarde uit een tabel, die bijvoorbeeld te vinden is in [5]. De berekening van het vorige voorbeeld is in de volgende tabel weergegeven. F(x) is de theoretische cumulatieve verdelingsfunctie, S(x) de waargenomen functie.

x	S(x)	F(x)	|F(x) - S(x)|
< 59,5	0.10390	0.085	0.0189
< 69,5	0.30519	0.353	0.0478
< 79,5	0.77273	0.730	0.0427
< 89,5	0.94156	0.946	0.0044
≦ 100,0	1.00000	1.000	0.0000

Bij $\alpha = 0.01$ is de kritische waarde $1.63/\sqrt{154} = 0.1313$. De maximale D uit de berekening is 0.0478. Daar $0.0478 < 0.1313$ kan de nulhypothese niet worden verworpen.

A4 CORRELATIE EN REGRESSIE-ANALYSE

Het correlatieprobleem is het vraagstuk van de samenhang tussen twee variabelen X en Y. Het regressieprobleem handelt over de vraag welke schatting van een variabele Y hoort bij een gegeven waarde van een variabele X.

Voor het schatten van de rechte (!) die het beste aansluit bij een verzameling van n waarnemingsparen*)

$$(X_1,Y_1),\ (X_2,Y_2),\ \ldots,\ (X_n,Y_n)$$

wordt veelal de methode van de kleinste kwadraten gebruikt. We geven een korte uiteenzetting van deze methode voor twee variabelen.

De te schatten lijn geven we aan door: $\hat{Y} = aX + b$, zodat bij elke X_i $(i = 1,\ldots,n)$ een schatting van Y, aan te geven door $\hat{Y}_i$, kan worden berekend. De verschillen $(Y_i - \hat{Y}_i)$ worden aangeduid met e_i.

Het kleinste kwadratencriterium houdt in, dat de waarden a en b zo gekozen dienen te worden dat

$$\Sigma e_i^2 = \Sigma(Y_i - \hat{Y}_i)^2 \qquad \text{minimaal is.}$$

Omdat elke schatting van Y op de geschatte rechte lijn ligt, geldt: $\hat{Y}_i = aX_i + b$. Substitutie van deze uitdrukking in de te minimaliseren uitdrukking geeft:

$$\Sigma e_i^2 = \Sigma(Y_i - aX_i - b)^2 = f(a,b) \quad .$$

Het minimum van deze uitdrukking wordt bepaald door de partiële afgeleiden gelijk nul te stellen.

Het is rekentechnisch eenvoudiger de variabelen in afwijking van hun gemiddelden aan te geven.

Dus: $y_i = Y_i - \overline{Y}$, $\hat{y}_i = \hat{Y}_i - \overline{\hat{Y}}$ en $x_i = X_i - \overline{X}_i$.

N.B. $\overline{Y} = \overline{\hat{Y}}$.

Door deze aanpassing kan de bovenstaande vergelijking ook worden geschreven als:

$$\hat{y}_i = a \cdot x_i \quad .$$

---

*) In de voorgaande paragrafen zijn de waarnemingsuitkomsten weergegeven door kleine letters (x_i). In deze en de volgende paragrafen gebruiken we hoofdletters (X_i en Y_i). De waarnemingsuitkomsten in afwijking van het gemiddelde geven we nu aan door kleine letters (x_i en y_i).

Daar $e_i = Y_i - \hat{Y}_i = y_i - \hat{y}_i$, is dus

$$\Sigma e_i^2 = \Sigma(Y_i - \hat{Y}_i)^2 = \Sigma(y_i - \hat{y}_i)^2 = \Sigma(y_i - ax_i)^2 \quad .$$

Nu is:

$$\frac{\partial}{\partial a}\Sigma(y_i - ax_i)^2 = \Sigma 2(-x_i)(y_i - ax_i) = 0$$

Of na herleiding:

$$a = \frac{\Sigma x_i y_i}{\Sigma x_i^2}$$

Als a eenmaal is berekend, kan de *regressieconstante* b worden berekend uit:

$$\overline{Y} - a\overline{X} = b \quad .$$

De enkelvoudige correlatie-coëfficient

De uitdrukking $\frac{\Sigma xy}{n - 1}$ wordt wel de covariantie van X en Y genoemd; als symbool hiervoor zullen we S_{xy} gebruiken. S_{xy} kan als een maatstaf voor de samenhang tussen de variabelen X en Y worden beschouwd.

Een belangrijk nadeel van de covariantie is evenwel, dat de grootte ervan wordt beïnvloed door de keuze van de eenheden waarin X en Y zijn uitgedrukt.

Een andere maatstaf, die overigens zeer nauw met de bovenstaande samenhangt is de enkelvoudige correlatiecoëfficiënt. Deze wordt als volgt gedefinieerd:

$$r_{xy} = \frac{xy}{\sqrt{\Sigma x^2 \cdot \Sigma y^2}} = \frac{S_{xy}}{S_x \cdot S_y} = a \cdot \frac{S_x}{S_y}$$

Zoals onmiddellijk is in te zien worden de eenheden waarin X en Y in de teller zijn uitgedrukt 'weggedeeld' door dezelfde eenheden waarin X en Y in de noemer zijn uitgedrukt, zodat de uitkomst een *onbenoemd* getal is. Een belangrijke eigenschap!

De betekenis van de enkelvoudige correlatiecoëfficiënt blijkt uit de volgende af te leiden formule:

$$\frac{S_e^2}{S_y^2} = (1 - r^2) \quad \longrightarrow \quad \frac{S_e}{S_y} = \sqrt{(1 - r^2)}$$

S_y kan worden beschouwd als de standaardafwijking van Y en is dus een maat voor de fluctuaties in de waarnemingsuitkomsten van deze variabele.

S_e is de standaardafwijking van de uitkomsten $e_i = (Y_i - \hat{Y}_i)$ $(i = 1,2,\ldots,n)$ en is dus een maat voor de fluctuaties in de verschillen tussen de waarnemingsuitkomsten van de variabele Y en van de geschatte Y ($\hat{Y}$). Men kan dus zeggen dat S_e de standaardafwijking is van dat deel van de fluctuaties van Y, dat niet verklaard is op grond van de betrekking, die tussen X en Y bestaat.

In de onderstaande tabel is voor een aantal waarden van r de verhouding $\frac{S_e}{S_y}$ gegeven.

r	0,000	0,100	0,200	0,300	0,400	0,500	0,600	0,700	0,800	0,900	$\frac{S_e}{S_y}$
$\frac{S_e}{S_y}$	1,000	0,995	0,980	0,954	0,917	0,866	0,800	0,714	0,600	0,436	r

De verklaring $\frac{S_e}{S_y} \times 100\%$ kan worden beschouwd als het percentage van de variaties in Y dat niet verklaard wordt door de variaties in X.

Een $r = 0,5$ houdt dan in, dat circa 87% van de fluctuaties van Y *niet* wordt verklaard op grond van de betrekking, die tussen X en Y wordt verondersteld.

De standaardfouten van a en b

Bij de bepaling van de regressielijn worden a en b berekend op grond van een steekproef van n waarnemingen. Welke conclusies kunnen op grond van deze schattingen met behulp van de statistiek worden getrokken t.a.v. de populatieparameters?

Hiertoe kunnen we de varianties van a en b berekenen. Op grond van een aantal veronderstellingen t.a.v. het toegepaste mathematische model voor de schatting van α en β (de populatieparameters) kunnen de volgende schatters worden afgeleid:

$$S_a^2 = \frac{S_e^2}{\Sigma x_i^2} \qquad S_b^2 = \frac{S_e^2 \; \Sigma X_i^2}{n \; \Sigma x_i^2}$$

waarin:

$$S_e^2 = \frac{\Sigma e_i^2}{n - 2}$$

Hierbij worden S_a en S_b aangeduid als de standaardfouten van de schatters.

Autocorrelatie

Hiervóór zijn twee maatstaven behandeld voor de beoordeling van de geschatte regressielijn: de correlatiecoëfficiënt r en de standaardfout. Een derde maatstaf heeft betrekking op de onafhankelijkheid van de storingstermen.

Deze veronderstellingen geven we aan door $E(\varepsilon_i, \varepsilon_j) = 0$ voor $i \neq j$. Als hieraan niet is voldaan zeggen we, dat er sprake is van autocorrelatie.

Als gevolg van autocorrelatie zijn de coëfficiënten a en b onbetrouwbaar. De standaardafwijkingen van de schattingen worden foutief geschat. Voor het onderzoek naar autocorrelatie wordt wel één van de volgende maatstaven gebruikt:

- de maatstaf van Durbin-Watson of
- de von Neumann-ratio (vNR).

De maatstaf van Durbin-Watson luidt:

$$d = \frac{\sum_{t=2}^{n} (e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$

Voor de berekening van de vNR wordt d met $\frac{n}{n-1}$ vermenigvuldigd, hetgeen voor grote waarden van n nauwelijks verschil oplevert.

Bij afwezigheid van autocorrelatie en onder de veronderstelling van een constante variantie zou de teller ongeveer gelijk zijn aan $2\,\Sigma e_t^2$, zodat in dat geval $d = 2$.

Als de uitkomst van d groter is dan 2 moet tot negatieve autocorrelatie worden geconcludeerd en anders tot positieve autocorrelatie.

Exacte significantieniveaus voor d bestaan niet. In sommige boeken zijn tabellen opgenomen met boven- en ondergrenzen voor d bij verschillende waarden van n en het aantal verklarende variabelen.

Beschikbare computerprogramma's voor de toepassing van de kleinste-kwadratenmethode geven de uitgevoerde berekeningen wel in de volgende vorm weer:

```
Y = 2,4X + 338,6          R = 0,987
    (30%)  (15%)        vNR = 1,97
```

Hiermee kan de regressievergelijking worden beoordeeld op:

1. autocorrelatie d.m.v. vNR;
2. nauwkeurigheid d.m.v. de uitkomsten tussen haakjes onder de coëfficiënten die de standaardfouten weergeven in procenten van de coëfficiënten;
3. lineaire samenhang d.m.v. R.

Hiervoor hebben we ons beziggehouden met de toepassing van de kleinste-kwadratenmethode op een model met één verklarende variabele. Als er sprake is van meer verklarende variabelen en/of meer vergelijkingen worden de schattingstechnieken ingewikkelder. In zo'n geval kan men het beste een econometrisch handboek, bijvoorbeeld [3] raadplegen.

A5 VARIANTIE-ANALYSE

Variantie-analyse toetst het verschil tussen r gemiddelden van aselecte steekproeven aan de hand van de varianties tussen die steekproeven. Variantie-analyse wordt vaak gebruikt bij het onderzoek naar onderlinge afhankelijkheid bij kwalitatieve factoren. Voorbeelden van studiegebieden zijn kwaliteitszorg, landbouwproeftechniek, marktanalyse. Wij beperken ons tot het effect van één factor ('one-way ANOVA').

Voorbeelden zijn: 'Wat is de invloed van bemesting op de gemiddelde opbrengst van een gewas?' of 'Welke invloed heeft de wijze van verpakken op de verkoop van een produkt in de levensmiddelenbranche?'.

De nulhypothese luidt dat de eventueel verwachte afhankelijkheid afwezig is; men zegt ook wel dat de populatie homogeen is ten aanzien van het te onderzoeken kenmerk. Uiteraard is een en ander gesteld binnen de betrouwbaarheidsgrenzen. Gewoonlijk neemt men 95%.

Men trekt r waarnemingsreeksen met steekproefomvang n_j $(j = 1,\ldots,r)$. Het aantal waarden van de factor bepaalt r. De steekproefomvang n_j van de waarnemingsreeks j kan voor verschillende j ongelijk van grootte zijn. De totale steekproefomvang $N = \sum_j n_j$.

De nulhypothese is dus de uitspraak dat er geen verschil is tussen de steekproeven (deze zijn dan afkomstig uit dezelfde populatie). In de variantie-analyse betekent dit onderzoek naar de steekproefvarianties, namelijk: zijn de varianties schattingen van dezelfde populatievariantie?

Vanuit de waarnemingen, die men aan de experimentele variabele doet, is het gemiddelde en de variantie te berekenen. Deze worden het totale gemiddelde ('grand mean') en de totale variantie genoemd. De totale variantie bestaat uit twee gedeelten.

1. De systematische variantie of experimentele variantie of 'tussenvariantie' ('between') is de variantie die wordt veroorzaakt door verschillen *tussen* de steekproefgemiddelden, met andere woorden door de verschillende waarden van de factor.

2. De toevallige variantie of 'error variantie' of 'binnenvariantie' ('within') komt voort uit verschillen *binnen* elke steekproef. In het experiment worden de verschillen door de omgeving (storing, ruis, onbeheersbare factoren) veroorzaakt, zoals door de aan de trekkingen ten grondslag liggende toevalsgeneratoren.

Als nu blijkt dat de tussenvariantie groter is dan de binnenvariantie, dan moet men erop verdacht zijn dat de steekproeven significant van elkaar verschillen: de nulhypothese wordt verworpen. Dit wordt met behulp van de F-toets onderzocht.

We slaan de waarnemingen in een tabel op:

steekproef	uitkomsten (waarnemingen)					totaal	gemiddelde	variantie
1	Y_{11}	Y_{12}	Y_{13}	...	Y_{1,n_1}	Tot_1	$Y_{1.}$	S_1^2
2	Y_{21}	Y_{22}	Y_{23}		Y_{2,n_2}	Tot_2	$Y_{2.}$	S_2^2
⋮	⋮	⋮	⋮		⋮	⋮	⋮	⋮
r	Y_{r1}	Y_{r2}	Y_{r3}		Y_{r,n_r}	Tot_r	$Y_{r.}$	S_r^2

Voor het totaal Tot_j van steekproef j geldt:

$$Tot_j = \sum_{k=1}^{n_j} Y_{jk}$$

Het gemiddelde $Y_{j\cdot} = Tot_j/n_j$

De variantie $S_j^2 = \sum_k (Y_{jk} - Y_{j\cdot})^2/(n_j - 1)$

Het algemeen totaal $\sum_j Tot_j$

De totale omvang $N = \sum_j n_j$

Het algemeen gemiddelde $Y_{\cdot\cdot} = totaal/N$

Er geldt voor een waarnemingsreeks, dat

$$Y_{jk} = \mu + T_j + \varepsilon_{jk}$$

Hierbij is μ het algemeen gemiddelde, met als schatter $Y_{\cdot\cdot}$. T_j is het effect van de betreffende waarde van de factor. Hieruit volgt dat T_j te schrijven is als:

$$T_j = (Y_{j\cdot} - Y_{\cdot\cdot}) \qquad \text{voor elke j.}$$

De storingsterm ε meet het verschil tussen de waarnemingen, zodat geldt:

$$\varepsilon_{jk} = (Y_{jk} - Y_{j\cdot}) \qquad \text{voor elke waarneming.}$$

Meestal wordt aangenomen, dat ε normaal verdeeld is, met:

$$E(\varepsilon_{jk}) = 0$$

$$var(\varepsilon_{jk}) = \sigma^2 \qquad \text{voor alle j,k}$$

We schrijven dit wel als:

$$\varepsilon_{jk} = N(0,\sigma^2) \quad .$$

Invulling van de gevonden uitdrukkingen levert de volgende vergelijking:

$$Y_{jk} = Y_{..} + (Y_{j.} - Y_{..}) + (Y_{jk} - Y_{j.})$$

anders geschreven:

$$(Y_{jk} - Y_{..}) = (Y_{j.} - Y_{..}) + (Y_{jk} - Y_{j.})$$

Kwadratering van beide leden en sommering over j en k geeft na herleiding de volgende vergelijking:

$$\sum_j \sum_k (Y_{jk} - Y_{..})^2 = \sum_j n_j (Y_{j.} - Y_{..})^2 + \sum_j \sum_k (Y_{jk} - Y_{.})^2$$

In woorden: De totale kwadraatsom(afwijking) = kwadraatsom ('tussen') + kwadraatsom('binnen'):

$$KS_{Tot} = KS_T + KS_B \quad .$$

Omdat onder de nulhypothese de verwachting van alle varianties $s_j^2 = \sigma^2$, geldt het volgende:

$$E(KS_B) = \sum_j (n_{j-1}) E(s_j^2) = (N-r)\sigma^2$$

N-r is het aantal vrijheidsgraden van KS_B.

$$E(KS_{Tot}) = (N-1)E(s^2) = (N-1)\sigma^2 \quad .$$

N-1 is het aantal vrijheidsgraden van de totale kwadraatsom.

Dus: $E(KS_T) = (n-1-N+r)\sigma^2 = (r-1)\sigma^2$

waarbij (r-1) het aantal vrijheidsgraden van KS_T voorstelt.

Nu volgt het vergelijken van de berekende F-ratio met de theoretische F.

$$F = \frac{KS_T/(r-1)}{KS_B/(N-r)} = \frac{KS_T}{KS_B} * \frac{N-r}{r-1} \quad .$$

A6 VOORSPELLINGSTECHNIEKEN

De voorspellingstechnieken kunnen in drie groepen worden verdeeld:

- *Kwalitatieve technieken*

  Deze worden vooral toegepast als kwantitatieve gegevens schaars zijn, bijvoorbeeld de introductie van een nieuw produkt.

- *Tijdreeksanalyse-technieken*

  Deze technieken trachten gebruik te maken van de regelmaat die in veel historische reeksen te vinden is; men spreekt wel van formele methoden.

- *Causale technieken*

  Deze verzameling van technieken tracht relaties te leggen tussen verschillende variabelen en formaliseert de relaties in een wiskundig model (zie bijvoorbeeld A4).

We behandelen in deze paragraaf enkele Tijdreeksanalyse-technieken

Een historische reeks (tijdreeks) is een cijferreeks die betrekking heeft op waarnemingen in een aantal opeenvolgende, even lange tijdperioden of op een aantal opeenvolgende tijdstippen met een gelijke tijdsafstand daartussen.

Uit deze omschrijving volgt, dat men te maken kan hebben met voorraadgrootheden en met stroomgrootheden. Voor de analyse is het van belang nadruk te leggen op het type grootheid, zoals het onderscheid tussen continue en discrete variabelen relevant kan zijn.

In een historische reeks is vaak een aantal golfbewegingen te onderscheiden:

- beweging op lange termijn:
  - trend
  - conjunctuurfluctuaties

- seizoenbewegingen;

- toevallige schommelingen.

Door de invloed van met name de laatste twee typen golfbewegingen is het moeilijk het verloop van de reeks voor de nabije toekomst te voorspellen. Dit probleem tracht men op te lossen door toepassing van een 'smoothing'-techniek: berekening van een voortschrijdend gemiddelde ('moving average'), 'exponential smoothing' of regressierekening met de tijd als verklarende variabele.

a. Het voortschrijdend gemiddelde

$$\underline{X}_t = a + \underline{e}_t \qquad (t = 1,2,\ldots,n)$$

$\underline{e}_t$: toevallige fluctuatie.

Over een zeer lange periode kan het gemiddelde van de toevallige fluctuaties als nul worden aangenomen, zodat:

$$\frac{\Sigma X_t}{n} = \underline{\bar{X}} = a \; .$$

De waarde van a kan evenwel als gevolg van seizoensinvloeden en/of conjunctuurinvloeden voortdurend fluctueren. Wenst men deze periodiciteit te analyseren, dan moet men niet middelen over de gehele tijdsperiode, maar over een periode waarin een volledige golfbeweging tot stand komt. Bij seizoensbewegingen bijvoorbeeld gemiddelden van 12 maanden of 4 kwartalen. Door te beginnen met de berekening van het gemiddelde van de eerste n waarnemingen en vervolgens van de 2° t/m (n+1)° waarneming, enzovoort, wordt een reeks getallen verkregen, die als 'voortschrijdende gemiddelden' worden aangeduid.

Naarmate n groter wordt gekozen, zal het effenend effect groter zijn. Een grote n zal een lage reactiesnelheid in de cijferreeks tot gevolg hebben, terwijl een lagere n wel een hogere reactiesnelheid met zich meebrengt, maar een vermindering van het 'smoothing' effect. De juiste keuze van n is niet altijd zo gemakkelijk als wel eens in de literatuur wordt voorgesteld.

Bij de techniek van het voortschrijdend gemiddelde wordt aan elke waarnemingsuitkomst in het verleden een even groot gewicht toegekend, voor zover de uitkomst althans in de berekening is opgenomen. In veel gevallen zal men evenwel aan recente informatie meer gewicht willen toekennen dan aan informatie uit een verder verleden. Om dit te realiseren kan een gewogen gemiddelde worden toegepast.

b. Exponential smoothing

In de door R.G. Brown ontwikkelde methode van het exponentieel effenen worden de gewichten voor de in het verleden verkregen waarnemingsuitkomsten gekozen volgens een meetkundige rij:

$$\alpha,\ \alpha(1-\alpha),\ \alpha(1-\alpha)^2,\ \ldots\ldots,\ \alpha(1-\alpha)^n$$

waarin:

$$0 < \alpha < 1$$

Als uitgegaan wordt van de waarnemingsuitkomsten $X_t, X_{t-1}, X_{t-2}, \ldots\ldots, X_{t-n}$ dan kan een voorspelling voor de periode (t+1) worden gedaan volgens de berekening:

$$\hat{X}_{t+1} = \alpha X_t + \alpha(1-\alpha)X_{t-1} + \alpha(1-\alpha)^2 X_{t-2} + \ldots\ldots\ldots\ldots\ldots$$

$$= \alpha X_t + (1-\alpha)[\alpha X_{t-1} + \alpha(1-\alpha)X_{t-2} + \ldots\ldots\ldots\ldots\ldots]$$

$$= \alpha X_t + (1-\alpha)\hat{X}_t \quad .$$

Door de keuze van α kan aan het recente verleden een meer of minder grote invloed worden toegekend. Een hoge waarde van α benadrukt de meest recente uitkomsten, terwijl een lagere α ook de uitkomsten in het verleden 'meeneemt'. Immers $\alpha = 0{,}9$ geeft als gewichten

0,9 0,09 0,009 0,0009 0,00009 ....

terwijl men bij $\alpha = 0{,}2$ de volgende reeks krijgt:

0,2 0,16 0,128 0,1024 0,082 ....

Een verlaging van α correspondeert met een verhoging van n bij de methode van de voortschrijdende gemiddelden. Beide methoden zijn gelijkwaardig als: $\alpha = \frac{2}{n+1}$.

Exponential smoothing is een populaire voorspellingstechniek door de eenvoud van het rekenwerk en de beperkte hoeveelheid gegevens die voor de berekening noodzakelijk is.

c. *Het lineaire regressiemodel*

Bij een positieve trend in de historische reeks verkrijgt men bij toepassing van beide voorgaande methoden voorspellingen, die voortdurend lager liggen dan de werkelijke uitkomsten. Als in de uitkomsten een lineaire trend aanwezig is, moet worden uitgegaan van het model:

$$\underline{X}_t = \alpha_0 + \alpha_1 t + \underline{e}_t$$

Voorspellingen voor dit model kunnen worden gedaan d.m.v. 'double exponential smoothing' of door toepassing van de kleinste-kwadraten-methode. Voor 'double exponential smoothing' wordt verwezen naar de literatuur, zie bijvoorbeeld [4] .

Bij toepassing van de methode der kleinste kwadraten wordt de tijd t als verklarende variabele beschouwd. Voor t worden waarden gekozen, die op gelijke afstand van elkaar liggen.

Bijvoorbeeld: Stel, dat de variabele X betrekking heeft op de jaren 1975 t/m 1987. Voor de te gebruiken t-waarden kunnen dan de waarden -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 en 6 worden gebruikt. Deze keuze vereenvoudigt de berekeningen, omdat het gemiddelde van de tijd nu nul is.
Als het aantal perioden of tijdstippen even is, kan worden gewerkt met de t-waarden:

------------, -7, -5, -3, -1, +1, 3, 5, 7, ------------ i.p.v.

------- , $-3\frac{1}{2}$, $-2\frac{1}{2}$, $-1\frac{1}{2}$, $-\frac{1}{2}$, $+\frac{1}{2}$, $1\frac{1}{2}$, $2\frac{1}{2}$, $3\frac{1}{2}$, ----------

De keuze heeft evenwel consequenties voor de uitkomst van de te berekenen parameter a_1, zodat deze na de berekening dient te worden gecorrigeerd.

Als de schatters α_0 en α_1 door a_0 respectievelijk a_1 worden aangeduid, is de formule voor de rechte lijn:

$$\hat{X}_t = a_0 + a_1 t \ ,$$

of met de variabelen in afwijking van hun gemiddelde:

$$\hat{x}_t = a_1 t \ , \quad \text{waarin:}$$

$$a_1 = \frac{\Sigma x_t t}{\Sigma t^2}$$

a_0 kan worden berekend uit:

$$a_0 = \bar{X} - a_1 \bar{t} = \bar{X} \ ,$$

omdat op grond van de hierboven aanbevolen keuze der t-waarden $\bar{t} = 0$.

De regressiecoëfficiënt a_1 is in tegenstelling tot de correlatiecoëfficiënt wel te beïnvloeden door de keuze der eenheden van de variabelen, zoals uit het voorbeeld op de volgende bladzijde blijkt. Dit voorbeeld is tevens bedoeld om de hiervoor bedoelde methode tot schatting van de lineaire trend toe te lichten.

Voorbeeld:

tijd	X_t	$x_t = X_t - \bar{X}$	t	2t	$x_t \cdot 2t$	t^2	$(2t)^2 = 4t^2$
1975	20	-23,2	$-4\frac{1}{2}$	-9	208,8	20,25	81
1976	22	-21,2	$-3\frac{1}{2}$	-7	148,4	12,25	49
1977	25	-18,2	$-2\frac{1}{2}$	-5	91,0	6,25	25
1978	40	- 3,2	$-1\frac{1}{2}$	-3	9,6	2,25	9
1979	42	- 1,2	$-\frac{1}{2}$	-1	1,2	0,25	1
1980	50	6,8	$+\frac{1}{2}$	1	6,8	0,25	1
1981	55	11,8	$1\frac{1}{2}$	3	35,4	2,25	9
1982	58	14,8	$2\frac{1}{2}$	5	74,0	6,26	25
1983	60	16,8	$3\frac{1}{2}$	7	117,6	12,25	49
1984	60	16,6	$4\frac{1}{2}$	9	151,2	20,25	81

$$\bar{X} = 43,2$$

$$\Sigma x_t \cdot 2t = 2\Sigma x_t \cdot t = 844,0 \rightarrow \Sigma x_t \cdot t = 422,0$$

$$\Sigma 4t^2 = 330 \rightarrow 4\Sigma t^2 = 330 \rightarrow \Sigma t^2 = 82,50$$

$$a_1 = \frac{422}{82,5} = 5,1 \qquad a_0 = \bar{X} = 43,2$$

Regressievergelijking:

$$\hat{X}_t = 43,2 + 5,1t \quad .$$

De hier beschreven methode kan worden toegepast voor zowel voorzichtige (!) korte termijnschattingen als voor de uitschakeling van een lineaire trend in het geval van toepassing van regressierekening.

A7 VARIANTIE REDUCTIE

Het is mogelijk om de variantie van het steekproefgemiddelde te verlagen door de steekproefomvang te doen toenemen. Dit is echter een dure methode, want de standaardfout is omgekeerd evenredig met n.

Toen simulatie nog in de kinderschoenen stond en de automatiseringsfaciliteiten niet opwogen tegen de huidige middelen, was het erg kostbaar en ook tijdrovend om de variabiliteit te verminderen door de steekproefomvang te laten toenemen. Er is zodoende nogal wat onderzoek gedaan om langs andere wegen de nauwkeurigheid van de resultaten bij dezelfde betrouwbaarheid en steekproefomvang te verhogen.

De methoden die hiervoor zijn ontwikkeld, worden aangeduid als *variantiereductietechnieken*. Een aantal van deze technieken is, onafhankelijk van de opkomst van simulatie reeds jaren geleden als onderdeel van de statistiek ontwikkeld. Men denke bijvoorbeeld aan de methode van de gelede (gestratificeerde) steekproef (Engels: stratified sample). Hierbij wordt de populatie in een aantal deelpopulaties (strata) verdeeld en uit elk van deze delen wordt vervolgens een aselecte steekproef getrokken. De strata worden niet willekeurig gekozen, maar bestaan reeds 'van nature', doordat het te onderzoeken kenmerk op verschillende wijze in de deelpopulaties voorkomt, bijvoorbeeld doordat het gemiddelde en/of de standaarddeviatie in de verschillende geledingen niet gelijk zijn. Voor een uitgebreide behandeling van deze methoden verwijzen we naar [2] .

Naast deze klassieke technieken zijn er in de afgelopen jaren diverse andere technieken onderzocht. Zo heeft Kleynen aan de variantiereductietechnieken veel aandacht besteed, zie bijvoorbeeld [6] . We bespreken een aantal van deze technieken.

a. Antithetische variabelen

Het principe van deze techniek is erop gericht een sterke negatieve correlatie tot stand te brengen tussen twee reeksen uitkomsten. Dit wordt bewerkstelligd door gebeurtenissen (door het toeval bepaald) te compenseren met gebeurtenissen met een tegengesteld effect. Lange helptijden worden gecompenseerd door korte helptijden; idem wachttijden, aankomsttijden, enzovoort. Men gebruikt hiervoor de toevalsgeneratorreeksen: elk toevalsgetal r wordt in het experiment nogmaals gebruikt, maar dan in de vorm (1-r). Als dus in de eerste 'run' wordt gewerkt met:

$$r_0,\ r_1,\ r_2,\ \ldots,\ r_n$$

dan wordt in een parallelle 'run' gewerkt met:

$$1-r_0,\ 1-r_1,\ 1-r_2,\ \ldots,\ 1-r_n$$

We krijgen dus, als we te maken hebben met een enkelvoudig experiment steeds twee uitkomsten (s_i en s_i^1) van dezelfde responsevariabele Y. Deze twee uitkomsten worden gemiddeld door te berekenen:

$$Y_i = \frac{s_i + s_i^1}{2} \qquad i = 1,2,\ldots,n .$$

De variantie van Y kan worden berekend volgens:

$$\text{var}(Y) = \{\text{var}(s) + \text{var}(s^1) + 2\ \text{cov}(s_1, s_1^1)\}/4$$

Daar de reeksen zodanig zijn gekozen, dat $\text{cov}(s_1, s_1^1)$ negatief is, zal var(Y) in dit geval kleiner zijn dan als gewerkt zou worden met een enkelvoudige reeks.

Verrassende resultaten, in de zin van daling der variantie zijn met behulp van deze techniek bereikt.

b. Gemeenschappelijke toevalscijfers (Engels: correlated sampling)

Deze techniek kan vooral worden gebruikt als het doel van het onderzoek is de vergelijking van verschillende beleidsvormen, strategieën. Dit betekent, dat in de verschillende simulatie-experimenten de omstandigheden die niet van invloed zijn op het beleid zoveel mogelijk identiek moeten zijn. Het spreekt vanzelf, dat de keuze van de toevalscijfers geen invloed mag hebben op de keuze van de beleidsvormen. Vandaar dat elke afzonderlijke run met dezelfde serie toevalscijfers wordt uitgevoerd.

Stel, men vergelijkt twee strategieën met als responsevariabelen X en Y. De vraag: welke strategie is beter, kan worden opgelost door van beide variabelen het gemiddelde te bepalen en vervolgens het verschil.

Stel:

$$Z_i = X_i - Y_i \qquad \text{voor elke } i.$$

Dan is:

$$\bar{Z} = \bar{X} - \bar{Y} \quad \text{en} \quad \text{var}(Z) = \text{var}(X) + \text{var}(Y) - 2\ \text{cov}(X,Y) .$$

Er bestaat dus als $\text{cov}(X,Y) \neq 0$ een verband tussen de variabelen X en Y. Als dit het geval is, is de t-toets in de gebruikelijke vorm niet alleen ongeoorloofd maar bovendien onverstandig. Het is onverstandig, omdat zoals uit de bovenstaande formule blijkt var (Z) wordt gereduceerd door positieve correlatie tussen X en Y. Dat X en Y positief gecorreleerd zijn is te verwachten op grond van het gebruik van dezelfde toevalscijfers.

Vanwege deze afhankelijkheid is verder toepassing van de t-toets in de gebruikelijke vorm niet geoorloofd. Men moet nu de

zogenaamde differentie-t-toets gebruiken. Betrouwbaarheids-intervallen voor het verschil tussen de gemiddelden kunnen worden berekend uit:

$$t_1 = (\bar{X}-\bar{Y}) + t_{\frac{1}{2}\alpha} \cdot s_z \sqrt{\frac{1}{n}} \quad \text{en} \quad t_2 = (\bar{X}-\bar{Y}) - t_{\frac{1}{2}\alpha} \cdot s_z \sqrt{\frac{1}{n}}$$

Hierin is s_Z de standaardafwijking van het verschil:

$$s_z = \sqrt{\frac{\Sigma\{(X_i-Y_i) - (\bar{X}-\bar{Y})\}^2}{n-1}}$$

en t heeft (n-1) graden van vrijheid.

Enkele eenvoudige voorbeelden waarbij de techniek van de gemeenschappelijke toevalscijfers kan worden toegepast, zijn:

- de keuze in een loketsysteem voor één of voor twee loketten;
- vergelijking van de kosten van twee of meer machines;
- vergelijking van verschillende bestelstrategieën bij een voorraadsysteem.

Het is duidelijk dat aan de eis van reproduceerbaarheid van de betrokken processen moet worden voldaan. Men kan bijvoorbeeld bij een loketsysteem niet volstaan met dezelfde startwaarde van de randomgenerator. Bij de situatie met twee loketten zal het helpen op een andere manier verlopen dan in de situatie met één loket. Het kan dan voorkomen dat een bepaald toevalsgetal, dat eerst voor een aankomsttijd werd gebruikt, nu voor een helptijd wordt gebruikt. De statistiek van de vergelijkingsresultaten is dan volstrekt onbetrouwbaar. De oplossing is dan om voor alle onderscheiden activiteitenreeksen verschillende reeksen toevalsgetallen te gebruiken.

Wat de praktische toepassing betreft, is het de moeite waard te overwegen of het mogelijk is i.p.v. twee experimenten na elkaar uit te voeren, één experiment te draaien waarbij de beide regels simultaan worden toegepast.

We hebben hierboven twee variantiereductietechnieken behandeld die elk in een bepaalde situatie, afhankelijk van het doel van de simulatie gebruikt kunnen worden. Over de voor- en nadelen zijn de meningen verdeeld, mede vanwege het beperkte onderzoek dat heeft plaats gehad.

A8 LITERATUUR

[1] W.G. Cochran, Sampling techniques, New York, 1953

[2] J.J.A. Moors en J. Muilwijk, Steekproeven, een inleiding tot de praktijk, Amsterdam, 1975

[3] H. Theil, Principles of Econometrics, New York, 1971

[4] W.G. Sullivan en W.W. Claycombe, Fundamentals of Forecasting, Reston, 1977

[5] W. Graybeal en U.W. Pooch, Simulation: principles and methods, Cambridge, 1980

[6] J.P.C. Kleynen, Statistical Techniques in Simulation, New York, 1974 (2 delen)

APPENDIX B

Wachtrijprocedures

B1 INLEIDING

In veel situaties waarin men met behulp van simulatie onderzoek wil doen, komen wachtrijproblemen voor.

Wanneer men SIMULA gebruikt, zal het voorkomen van wachtrijen weinig problemen opleveren; de taal beschikt immers over een aantal standaardprocedures waarmee men snel en gemakkelijk rijen kan definiëren en hanteren.

Niet iedereen echter kan of wil met SIMULA werken. Wanneer men de voorkeur geeft aan proceduregerichte talen als Pascal of BASIC zal men zelf wachtrijprocedures moeten ontwikkelen. Wij geven hier een *voorbeeld* van een mogelijke oplossing voor het wachtrijprobleem in BASIC en in Pascal. Hierbij wordt aangetekend, dat het niet de bedoeling is om een toepassing van *simulatie* te laten zien; men zal dus de statistische procedures en de kenmerken van een experimenteel ontwerp vergeefs zoeken in de voorbeeldprogramma's. Voor de specifieke eigen toepassingen zal men zelf de nodige aanpassingen hieromtrent moeten inbrengen.

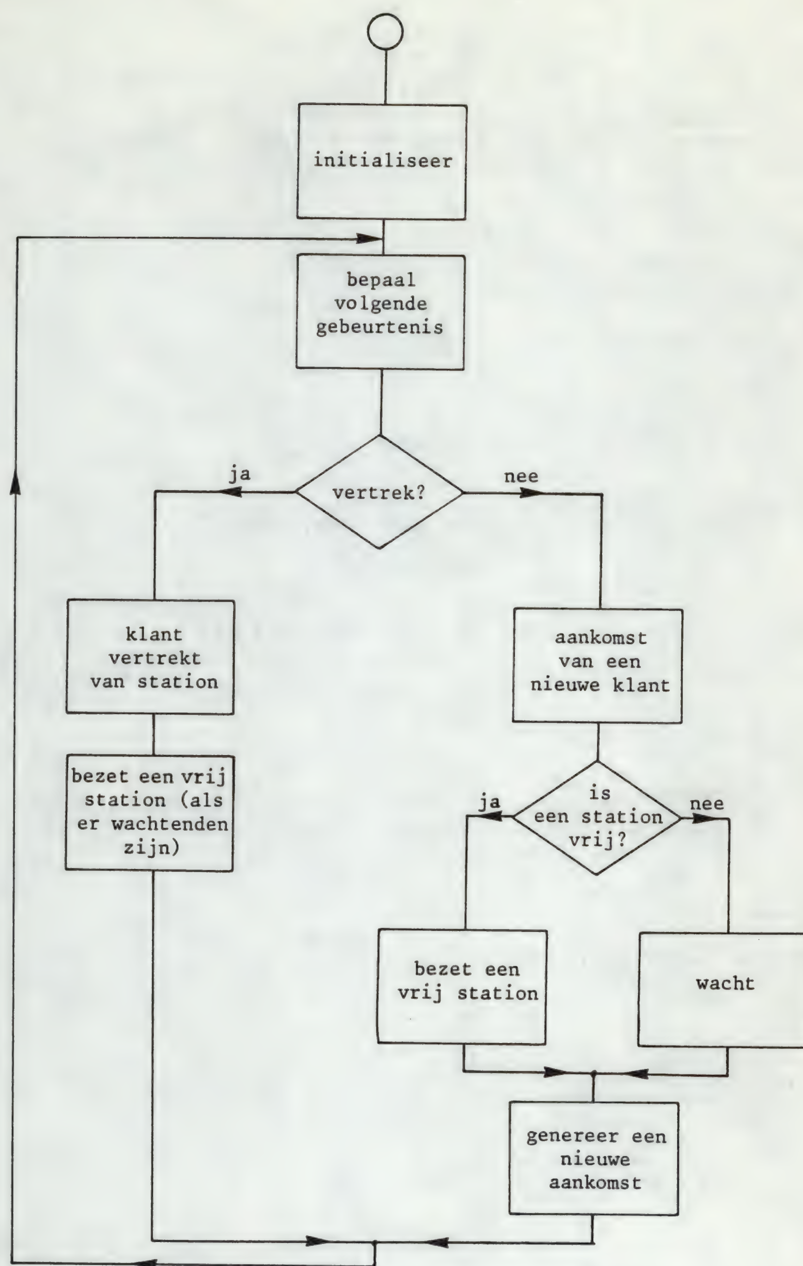
B2 EEN WACHTRIJSIMULATIE IN MICROSOFT-BASIC

Het volgende programma betreft een systeem waarbij klanten aankomen volgens een Poisson proces. Er is één wachtrij, maar er kunnen meer bedieningsstations zijn, waar de klanten worden geholpen. De helptijd is uniform verdeeld.

Het programma stelt enige vragen, alvorens de simulatie een aanvang neemt. Opgegeven moet worden:

- de gemiddelde tussenaankomsttijd;
- de minimale helptijd;
- de maximale helptijd;
- het aantal bedieningsstations (maximum aantal: 10);
- het aantal te verwerken klanten.

Het proces wordt weergegeven in het programmastroomschema.



Structuur van het programma

```
10      REM SIMULATIE VAN WACHTRIJEN
20      'initialiseer
30      DIM TST$(10), VERTREK(10), HELP(10)
40      PRINT CHR$(27) + "[2J"   'schoon scherm
50      INPUT "wat is de gemiddelde tussenaankomsttijd?  ";TUSTIJD
60      INPUT "wat is de minimale helptijd?  ";MIN
70      INPUT "wat is de maximale helptijd?  ";MAX
80      INPUT "hoeveel bedieningsstations zijn in gebruik?  ";AST%
90      INPUT "hoeveel klanten worden in het systeem gebracht?  ";AANTAL%
100     '
110     FOR S = 1 TO AST%
120         TST$(S) = "vrij"
130     NEXT S
140     'eerste aankomst
150     TIJD = 0
160     F = -LOG(1-RND)
170     F = INT(F)
180     TAK = F * TUSTIJD
190     TIJD = TAK
200     RIJLENGTE = 1
210     EVSTAT = 1
220     GOSUB 3500  'bezet station
230     GOSUB 4000'genereer volgende aankomst
240     '
250     'hoofdprogramma
260     WHILE TOTHELP <= AANTAL%
270            GOSUB 1000  'bepaal de volgende gebeurtenis
280            IF WISSEL$ = "vertrek"
                  THEN GOSUB 2000 : GOSUB 3000
                  ELSE GOSUB 4000
290            '2000=vertrek; 3000=bezet vrij station; 4000=aankomst
300            INPUT "druk op return om verder te gaan"; A$
310     WEND
320     END
330     '
340     '
1000    'bepaal de volgende gebeurtenis
1010    TIJDSPR = TAK - TIJD
1020    WISSEL$ = "aankomst"
1030    FOR S = 1 TO AST%
1040         IF TST$(S) = "bezet"
                THEN IF TIJDSPR >= VERTREK(S) - TIJD
                          THEN GOSUB 1200
1050    NEXT S
1060    TIJD = TIJD + TIJDSPR
1070    RETURN
1080    '
1200    TIJDSPR = VERTREK(S) - TIJD
1210    WISSEL$ = "vertrek"
1220    EVSTAT = S
1230    RETURN
1240    '
2000    'vertrek
2010    TST$(EVSTAT) = "vrij"
2020    TOTBEZSTAT = TOTBEZSTAT - 1
2030    RIJLENGTE = RIJLENGTE - 1
2040    TOTHELP = TOTHELP + 1
2050    HELP(EVSTAT) = HELP(EVSTAT) + 1
2060    GOSUB 6000 ' druk1
2070    RETURN
2080
```

```
3000    'bezet vrij station
3010    IF RIJLENGTE - TOTBEZSTAT > 0
           THEN GOSUB 3100
3020    RETURN
3100    FOR S = 1 TO AST%
3110        IF TST$(S) = "vrij"
               THEN EVSTAT = S: GOSUB 3500: S = AST%
3120    NEXT S
3130    RETURN
3140    '
3500    'bezet station
3510    TST$(EVSTAT) = "bezet"
3520    TOTBEZSTAT = TOTBEZSTAT + 1
3530    VERTREK(EVSTAT) = (MAX - MIN + 1) * RND + MIN + TIJD
3540    VERTREK(EVSTAT) = INT(VERTREK(EVSTAT))
3550    GOSUB 6200  'druk2
3560    RETURN
3570    '
4000    'aankomst
4010    RIJLENGTE = RIJLENGTE + 1
4020    GOSUB 3000  ' bezet vrij station
4030    GOSUB 4200  ' genereer volgende aankomst
4040    GOSUB 6400  ' druk3
4050    RETURN
4060    '
4200    'genereer volgende aankomst
4210    F = -LOG(1 - RND)
4220    F = INT(F)
4230    TAK = TIJD + F * TUSTIJD
4240    RETURN
4250    '
6000    'druk1
6010    PRINT "vertrek"
6020    PRINT "de tijd is  ";TIJD
6030    PRINT "het totaal aantal geholpen klanten is: ";TOTHELP
6040    PRINT "het stationnummer ";EVSTAT; " is  vrij"
6050    PRINT "het aantal geholpen klanten bij station ";
        EVSTAT;" is ";HELP(EVSTAT)
6060    RETURN
6070    '
6200    'druk2
6210    PRINT "bezet"
6220    PRINT "de tijd is ";TIJD
6230    PRINT "het station nummer ";EVSTAT; " is bezet."
6240    PRINT "de vertrektijd bij station ";EVSTAT; "is ";VERTREK(EVSTAT)
6250    RETURN
6260    '
6400    'druk3
6410    PRINT "aankomst"
6420    PRINT "de tijd is ";TIJD
6430    PRINT "de aankomsttijd is ";TAK
6440    PRINT "de rijlengte is ";RIJLENGTE
6450    RETURN
6460    '
```

Opmerkingen

1. De uitvoer loopt door totdat het aantal geholpen klanten één meer is dan in de invoer is opgegeven.

2. De verdelingsfuncties (regel 160, 3530 en 4210) kan men zelf wijzigen in andere functies.

3. De 'integer' functies op de regels 170, 3540 en 4220 maken van de getrokken aankomst- en bedieningstijd gehele getallen: men dient voor experimentatie deze regels te verwijderen.

4. Er is geen beveiliging tegen onjuiste of foutieve invoer.

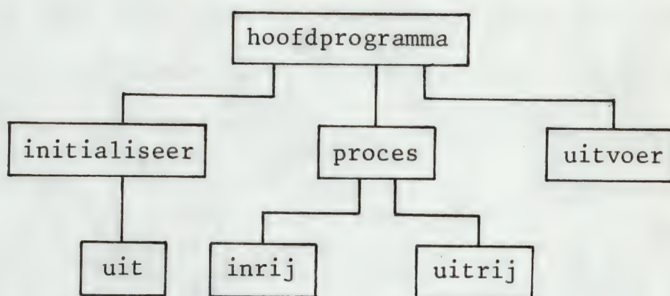
5. De betekenis van enige namen:

   AST%: aantal stations
   TAK: tussenaankomsttijd
   EVSTAT: eerst volgend station
   TIJDSPR: tijdsprong
   TST$(S): toestand van station S
   TOTBEZSTAT: het totaal aantal bezette stations
   HELP(S): het aantal geholpen klanten bij station S

B3 EEN WACHTRIJSIMULATIE IN PASCAL (een loketsysteem)

I De structuur van het totale programma

In In dit programma wordt een systeem gesimuleerd, waarbij één wachtrij en één bedieningssysteem is betrokken. Voor de rijprocedures wordt gebruik gemaakt van het pointer-mechanisme. Er zijn géén verdelingsfuncties gebruikt, zodat men deze bij het gebruik van dit programma zelf moet definiëren. In plaats daarvan worden per klant het aankomsttijdstip en de bedieningstijd ingevoerd.



II Het hoofdprogramma

Uitvoer	- uitklanten.klantno:	het volgnummer van de klant.
	- tussenaank:	het verschil tussen de aankomsttijd van de huidige klant en de aankomsttijd van de vorige klant.
	- uitklanten.aankomsttijd:	tijdstip van aankomst.
	- uitklanten.bedieningstijd:	tijdsduur waarin de klant aan het loket wordt geholpen.
	- vertrek:	tijdstip van vertrek.
	- wachttijd:	verblijfsduur bij het loket = vertrektijd minus de aankomsttijd.

Invoer Per klant worden de volgende gegevens ingevoerd:

- klanten.aankomsttijd:	tijdstip van aankomst.
- klanten.bedieningstijd:	tijdsduur waarbinnen de klant aan het loket wordt geholpen.

Proces Voor een onbekend aantal klanten moeten de volgende zaken gesimuleerd worden:
- de aankomst van een klant
- het wachten van een klant
- het geholpen worden van een klant
- het vertrek van een klant

III Procedure initialiseren

Probleemdefinitie:

Uitvoer	- klanten.klant	dit record bevat in het veld klanten. aankomsttijd de aankomsttijd van de eerste klant
	- volgno	bevat het volgnummer van de eerste klant, is dus 1
	- wrkop:point	adresverwijzing naar het voorste record van de wachtrij
	- wrstaart:point	adresverwijzing naar het laatste record van de wachtrij
	- uitkop:point	adresverwijzing naar het voorste record van de uitvoerrij
	- uitstaart:point	adresverwijzing naar het laatste record van de uitvoerrij

Communicatie via de terminal:
uitleg over de te volgen procedure bij het invoeren van de gegevens en de vraag om de aankomsttijd van klant één in te voeren.

Invoer Er is geen programma-invoer maar wel invoer via de terminal:
klanten.aankomsttijd

Proces Initialiseren van de wachtrij en de uitvoerrij en bepalen wanneer de eerste klant aankomt.

Probleemanalyse:

1) Initialiseer de wachtrij, waarin de op hun beurt wachtende klanten worden opgeslagen
2) Initialiseer de uitvoerrij, waarin de klanten die vertrokken zijn worden opgeslagen
3) Geef uitleg over de manier van invoeren
4) Vragen om invoer van de aankomsttijd van de eerste klant en het invoeren van deze tijd.

Probleemoplossing:

init (wrkop,wrstaart)
init (uitkop,uitstaart)
schrijf uitleg
volgnummer wordt 1
vraag aankomsttijd van klant 1
lees (klanten.aankomsttijd)

IV Procedure proces

Probleemdefinitie:

Uitvoer	- uitstaart	:	point,	adresverwijzing naar het laatste record in de uitvoerrij. Deze vormt samen met uitkop de uitvoerrij.
Invoer	- klanten	:	klant,	dit record bevat in het veld klanten.aankomsttijd de aankomsttijd van de eerste klant.
	- volgno	:	int,	bevat het volgnummer van de eerste klant.
	- wrkop	:	point,	adresverwijzing naar het eerste record in de wachtrij.
	- wrstaart	:	point,	adresverwijzing naar het laatste record in de wachtrij.
	- uitstaart	:	point,	adresverwijzing naar het laatste record in de uitvoerrij.

Proces Het simuleren van het aankomen, wachten en verwerken van klanten.

uitvoer via terminal	:	vraag om bedieningstijd van een klant in te voeren.
invoer via terminal	:	klantenbedieningstijd = tijdsduur waarin de klant aan het loket wordt geholpen.
uitvoer via terminal	:	vraag om aankomsttijd volgende klant in te voeren.
invoer via terminal	:	klantenaankomsttijd.

Lokale parameters	:	- tijd	Teller die de tijd bijhoudt. Deze wordt elke cyclus met 1 verhoogd.
		- vertrektijd	tijdstip waarop de klant aan het loket klaar is.
		- loket	geeft aan of het loket al dan niet bezet is: true: loket is niet bezet false: loket is bezet
		- loketklant	bevat alle tot nu toe verkregen informatie over de klant die aan het loket staat.

Probleemanalyse:

```
Zolang er nog klanten aankomen of de wachtrij niet leeg is of er nog
                    nog iemand aan het loket staat

    Zolang er op dit tijdstip klanten arriveren
        voer hun bedieningstijd in
        geef ze een volgnummer
        zet ze in de wachtrij
        bepaal wanneer de volgende klant komt

    Als de klant is geholpen (vertrektijd = tijd)
        zet de klant in de uitvoerrij
        geef het loket vrij

    Als het loket leeg is en er bevindt zich iemand in de wachtrij
        haal de klant uit de wachtrij
        bezet het loket
        stel het begin van de bedieningstijd vast
        bepaal de vertrektijd

    Verzet de tijd 1 eenheid
```

Probleemoplossing:

```
tijd := 0   ;   loket := leeg

klanten.aankomsttijd ≠ -1 of wrkop ≠ wrstaart of loket ≠ leeg
    klanten.aankomsttijd = tijd
        writeln('bedieningstijd klant',volgno)
        readln(klanten.bedieningstijd)
        klanten.klantno = volgno
        intij(klanten,wrstaart)
        volgno = volgno + 1
        writeln('aankomsttijd klant',volgno)
        readln(klanten.aankomsttijd)
    vertrektijd = tijd
        true:
            inrij(loketklant,uitstaart)
            loket = leeg
        false:
    loket = leeg en wrkop ≠ wrstaart
        true:
            uitrij(loketklant,wrkop)
            loket = niet leeg
            loketklant.beginbedieningstijd = tijd
            vertrektijd = tijd + loketklant.bedieningstijd
        false:
    tijd = tijd + 1
```

V Beschrijving van de globale variabelen

Constante beschrijving:

leeg : boolean, met de waarde true; de variabele loket wordt aan deze constante getoetst om te bepalen of er mensen aan het loket worden geholpen.

Type beschrijvingen:

klant : record, bevat een aantal velden die relevante gegevens bevatten.

Deze velden zijn:
- klantno: integer, volgnummer van de klant
- aankomsttijd: integer, tijdstip van aankomst
- bedieningstijd: integer, tijdsduur waarin de klant aan het loket wordt geholpen
- bedieningstijd: integer, tijdstip waarop de klant aan de beurt is.

point : <u>pointertype</u>, bevat het adres van het bijbehorende record.

rec : <u>record</u>, het record dat bij point hoort; het bevat de volgende velden:
- kl: klant, record dat een aantal relevante gegevens over de klanten bevat
- volgende: point, wijzer (pointer) naar het volgende record.

VI Opmerkingen

Aanname ten aanzien van het model:
- het proces wordt gestopt als de laatste klant vertrekt

Aannamen ten aanzien van het programma:
- er vindt geen controle plaats op syntaxfouten of semantische fouten bij de invoergegevens.
- de tijdseenheid waarmee het programma werkt, wordt niet nader gespecificeerd.

VII Het Pascalprogramma

```
program loket (input, output);

const leeg = true;

type klant = record
                klantno                   ,
                aankomsttijd              ,
                bedieningstijd            ,
                beginbedieningstijd       : integer;
             end;

     point = ^rec;

     rec = record
             kl          : klant;
             volgende    : point;
           end;

 var wrkop        ,
     wrstaart     ,
     uitkop       ,
     uitstaart    : point;
     volgnr       : integer;
     clienten     : klant;

procedure init (var kop, staart: point);

begin (*init*)
      new (kop);
      staart := kop;
end;

procedure initialiseer (var klanten  : klant;
                        var volgno   : integer;
                        var wrkop, wrstaart, uitkop, uitstaart: point);
    begin
        init (wrkop, wrstaart);
        init (uitkop, uitstaart);
        writeln ('invoergegevens voor de aankomende klanten:');
        writeln ('**aankomsttijd (deze tijd dient als "klok")');
        writeln ('**bedieningstijd');
        writeln ('om het programma te beeindigen voert U "-1" in');
        writeln;
        volgno := 1;
        write ('aankomsttijd klant ',volgno,'                    :  ');
        readln (klanten.aankomsttijd);
     end; (*initialiseer*)

procedure inrij (moetinrij: klant; var staart: point);
    begin
        staart^.kl := moetinrij;
        new (staart^.volgende);
        staart := staart^.volgende;
    end; (*inrij*)

procedure uitrij (var komtuitrij: klant; var kop: point);
    var actueel: point;
```

```
    begin
        actueel := kop;
        kop := kop^.volgende;
        komtuitrij := actueel^.kl;
    end; (*uitrij*)

 procedure proces (klanten: klant; volgno: integer;
                   wrkop, wrstaart: point; var uitstaart: point);
     var tijd, vertrektijd: integer; loket: boolean; loketklant: klant;
     begin
         tijd := 0;
         loket := leeg;
         while (klanten.aankomsttijd <> -1) or (wrkop <> wrstaart) or
               (loket = not leeg) do
             begin
                 while klanten.aankomsttijd = tijd do
                     begin
                         write ('bedieningstijd klant ',
                         volgno,'                 :  ');
                         readln (klanten.bedieningstijd);
                         klanten.klantno := volgno;
                         inrij (klanten,wrstaart);
                         writeln;
                         writeln;
                         volgno := volgno + 1;
                         write ('aankomsttijd klant ',volgno,
                         ' (of "-1")        : ');
                         readln (klanten.aankomsttijd);
                      end;
                      if vertrektijd = tijd
                         then begin
                                  inrij (loketklant, uitstaart);
                                  loket := leeg;
                              end;
                      if (loket = leeg) and (wrkop <> wrstaart)
                         then begin
                                  uitrij (loketklant,wrkop);
                                  loket := not leeg;
                                  loketklant.beginbedieningstijd := tijd;
                                  vertrektijd := tijd +
                                                 loketklant.bedieningstijd;
                              end;
             tijd := tijd + 1;
             end;
    end; (*proces*)

procedure uitvoer (uitkop, uitstaart : point);
    var vorigeaank, tussenaank, vertrek, wachttijd: integer;
        uitklanten: klant;
    begin
        writeln;
        writeln;
        writeln ('_____________________________________________________________________________');
        writeln ('| klant  tussen-  tijdstip  bedienings-  tijdstip  wacht- |');
        writeln ('|        aankomst van aan-     tijd         van      tijd  |');
        writeln ('|        tijd     komst                   vertrek          |');
        writeln ('_____________________________________________________________________________');
```

```
        vorigeaank := 0;
        while uitkop <> uitstaart do
            begin
                uitrij (uitklanten, uitkop);
                tussenaank := uitklanten.aankomsttijd - vorigeaank;
                vorigeaank := uitklanten.aankomsttijd;
                vertrek    := uitklanten.beginbedieningstijd +
                              uitklanten.bedieningstijd;
                wachttijd  := vertrek - uitklanten.aankomsttijd;
                write ('|',uitklanten.klantno:4,tussenaank:8,
                           uitklanten.aankomsttijd:9);
                writeln (uitklanten.bedieningstijd:11,
                         vertrek:12,wachttijd:10,'   |');
            end;
            writeln('________________________________________________________
    end; (*uitvoer*)

begin (*hoofdprogramma*)
    initialiseer (clienten, volgnr, wrkop, wrstaart, uitkop, uitstaart);
    proces (clienten, volgnr, wrkop, wrstaart,uitstaart);
    uitvoer (uitkop, uitstaart);
end.
```

Index

F

G

H

I

K

L

M

N

O

P

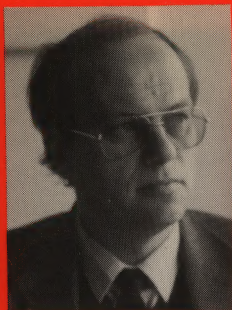
R

T

U

V

DE AUTEURS



JACQUES BOERSMA studeerde kwantitatieve bedrijfseconomie aan de RU te Groningen. Na een aantal jaren praktijkervaring te hebben opgedaan bij de AMRO-bank was hij docent in o.a. systeemanalyse en -ontwerp aan de heao te Den Haag. Van 1975-1977 was hij tevens voorzitter van het landelijke bedrijfsinformatica-ontwikkelteam (BIO-team) ten behoeve van de afdelingen bedrijfsinformatica bij het heao. Sinds 1977 is hij wetenschappelijk hoofdmedewerker Bestuurlijke Informatiekunde aan de Interfaculteit Bedrijfskunde van de RU-Groningen. Daarnaast adviseert hij organisaties over de ontwikkeling en invoering van geautomatiseerde informatiesystemen. Behalve dit boek heeft hij verschillende publikaties over Bestuurlijke Informatiekunde en Informatiebeleid op zijn naam staan.



THEO HOENDERKAMP studeerde aan de Technische Hogeschool te Delft wiskunde en informatica. Hij is vanaf de oprichting van de Bedrijfskundige Informatica Opleiding bij het heao (hoger economisch en administratief onderwijs) te Den Haag werkzaam als afdelingsvoorzitter en als docent Informatica, Informatiesystemen en Simulatie bij deze opleiding. De BI-opleiding beoogt een opleiding tot systeemontwerper en informatieanalist van administratieve en bestuurlijke informatiesystemen. Hoenderkamp heeft zich intensief beziggehouden met leerplan- en leerstofontwikkeling in het landelijke bedrijfsinformatica-ontwikkelteam (BIO-team). Hij is momenteel voorzitter van dit team. Daarnaast adviseert hij diverse instanties op het gebied van het informatica-onderwijs. Hij publiceerde in een aantal vakbladen over de ontwikkelingen in het bedrijfsinformatica- onderwijs.

HET BOEK

SIMULATIE is een onderzoekmethode met schier onbeperkte mogelijkheden. In deze tijd van micro's, mini's en supercomputers kan de methode door vrijwel een ieder worden gebruikt ter ondersteuning van beslissingen. Met behulp van simulatie is kennisvermeerdering mogelijk, zowel wat betreft de structuur van problemen als ten aanzien van de oplossingen hiervan. In de toekomstige ontwikkelingen van Decision Support Systems en Expertsystemen zal simulatie door de genoemde mogelijkheden een belangrijke rol gaan spelen. Om al deze redenen is het de moeite waard kennis te nemen van Simulatie.
Het boek wordt sinds een aantal jaren gebruikt in opleidingen voor management, systeemontwerpers, bedrijfseconomen en informatica. Geleidelijk worden de lezers ingevoerd in het vakgebied, zowel theoretisch als praktisch. In het boek zijn ter toelichting twee uitgewerkte cases en programma's in BASIC, Pascal en SIMULA opgenomen. Ook voor zelfstudie is het boek geschikt. De toegenomen technische mogelijkheden van PC's en microcomputers brengen simulatie binnen ieders bereik.